

# Modeling, Simulation and Analysis of Integrated Building Energy and Control Systems

Michael Wetter

Simulation Research Group

Building Technologies Department

Energy and Environmental Technologies Division

Lawrence Berkeley National Laboratory

August 10, 2009



# Overview

- Introduction

Trends - Problems - Needs

- Mono-Simulation with Modelica

Modelica Standard Library - LBNL Buildings Library - Applications

- Co-Simulation with Building Controls Virtual Test Bed

Analysis - Building Controls Virtual Test Bed - Applications

- R&D Needs

# Integration to Increase Efficiency

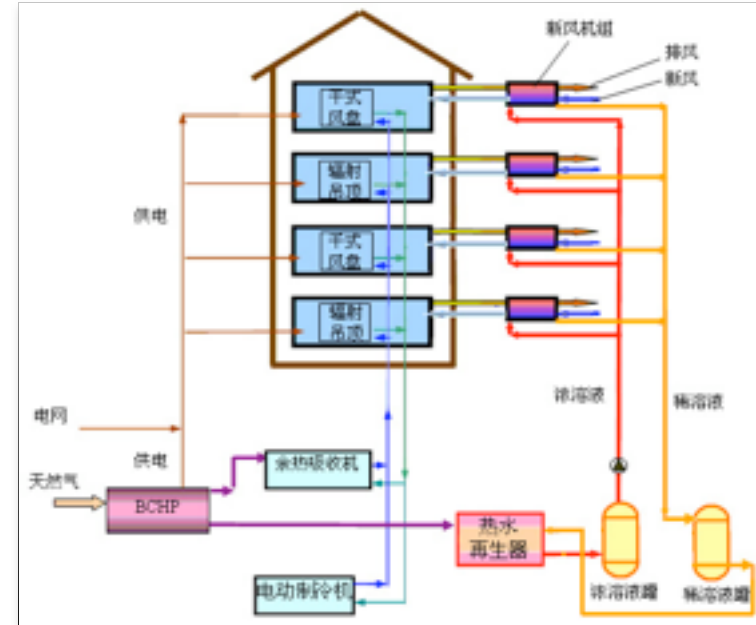
Active facade for natural ventilation



Micro-mirror to redirect sunlight



Decentralized dehumidification with liquid desiccant



Phase change material to increase thermal storage



Cyprus grass to humidify supply air



Web-server at the size of 25 cents



# Trends

Innovation happens at the interface between disciplines.

Integrated systems require system-level analysis.

Computational Science and Engineering reduces cost and time,  
*but* needs flexible tools

- for rapid prototyping
- to identify and fix mistakes early

New opportunities through C<sup>3</sup>:

Communication Computation Controls

# Smoothness of Simulation Output

Small changes in input  $x$  should cause small changes in output  $f(x)$ .

$$\frac{\partial f(x)}{\partial x^i} \approx \frac{f(x + \Delta e_i) - f(x)}{\Delta}$$

Smooth results are required for

controls

feedback linearization

state linearization

optimal control

optimization

nonlinear programming

pattern search methods

analysis

sensitivity

robustness

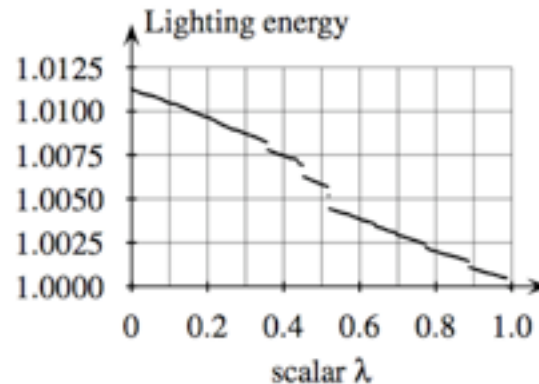
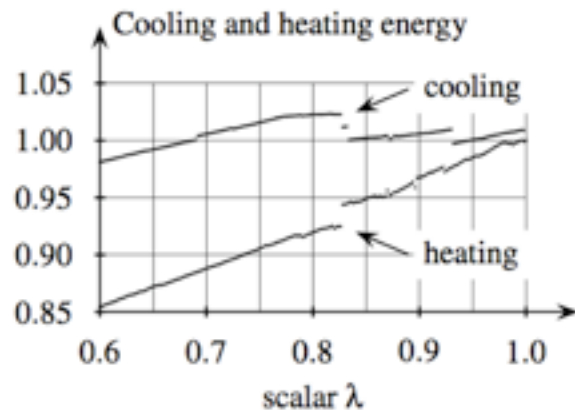
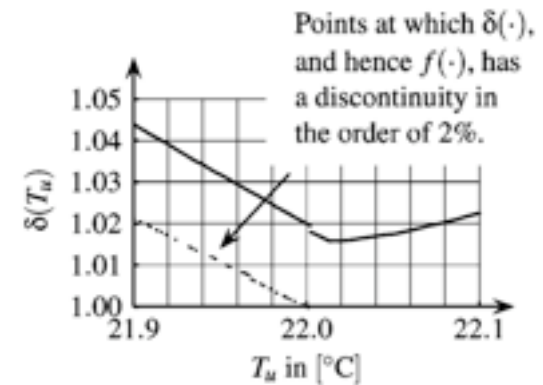
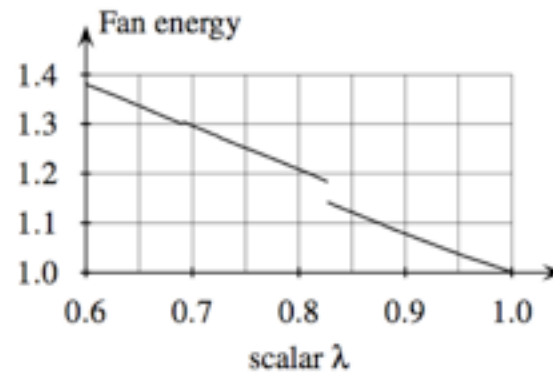
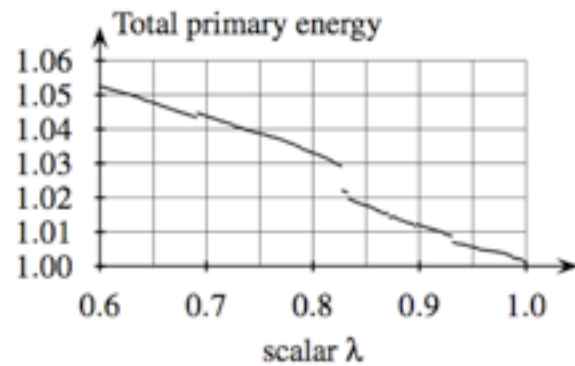
numerical solvers

Newton-based solvers

integration algorithm for stiff systems

# Numerical “Noise” in EnergyPlus

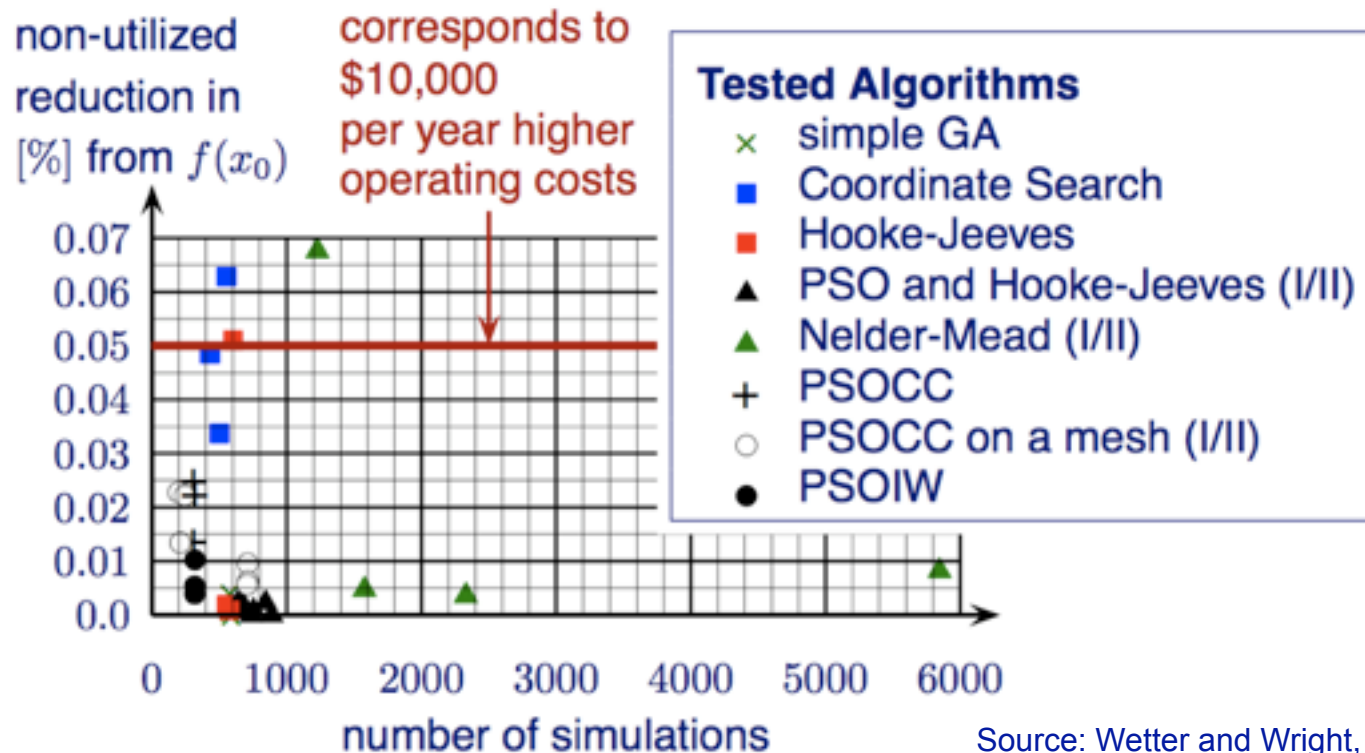
$$x(\lambda) = x_{HJ} + \lambda(x_{sGA} - x_{HJ}),$$
$$x \in \mathbb{R}^{13}$$



Discontinuities are caused by nested solvers with low precision.  
(20+ nested solvers spread over 500,000 lines of code.)

# Numerical “Noise” in EnergyPlus

Failure if EnergyPlus is used to approximate  $f(\cdot)$



Source: Wetter and Wright, 2004

Automated analysis is not necessarily robust with existing simulation programs.

Do we need to rethink how we develop simulation programs?

# Controls Oriented Modeling

## Needs

- Feedback control of states (temperature, pressure) not heating/cooling load
- Code generation for control hardware
- Freely programmable control sequences
  - graphical block diagrams
  - textual algorithms
  - hierarchies to manage complexity and encapsulate functional blocks
- Different models of computations
  - continuous time
  - discrete time
  - finite state machine
- Application programming interfaces (API) to tools used by controls engineers
- Analysis capabilities
  - linearization
  - state initialization
  - applicable for optimal control algorithms

# Parallel Computing

## New situation

- Hardware becomes parallel, CPU speed stagnates.
- Floating point operation is cheap, memory access is expensive.

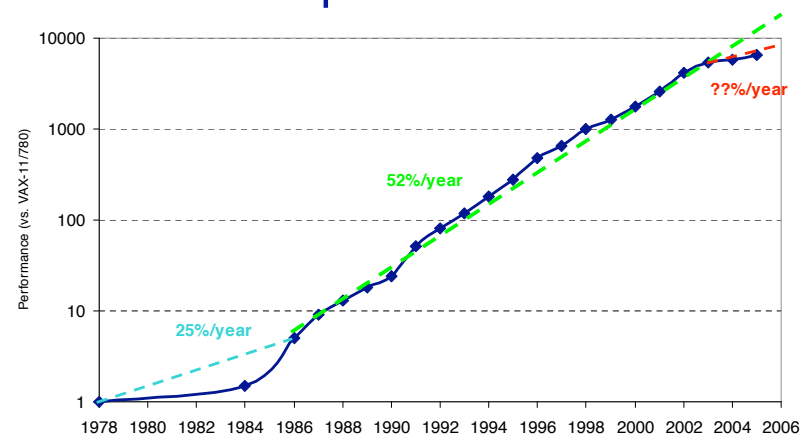
## EnergyPlus

- Rewrite 500,000 lines of code?
  - race conditions
  - memory management...
- Very expensive proposition.
- No formal verification possible.

## Equation-based languages

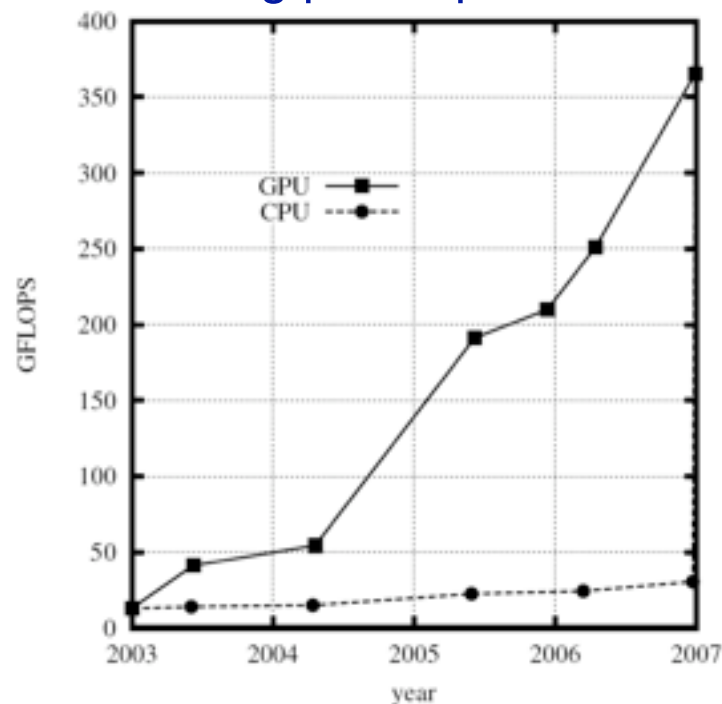
- Write analyzable code.
- Use language constructs to map subsystem models to processors.
- Change code generators to map strongly coupled equation systems to processors.

## Processor speed



Source: Asanovic et al., 2006

## Floating point operations



Source: Zuo and Chen, 2009

# Modeling of Physical Systems

Higher-level of abstraction to

- increase productivity
- facilitate model-reuse
- preserve system topology
- enable analysis
- generate code for target hardware

Procedural modeling  $\approx 1970$

```

program euler
  implicit none
  double precision, parameter :: tFin = 2000  ! Final time
  integer, parameter :: N = 20000            ! Number of steps
  integer, parameter :: hCon = 10            ! Communication counter
  integer :: iCon

  double precision dt                          ! Time step
  double precision, parameter :: T0 = 293.15  ! Initial temp.
  double precision :: T1, T2                 ! Temperature
  double precision :: TBC                     ! Temp. boundary condition
  double precision :: TSur                    ! Surface temperature
  double precision, parameter :: TSet = 293.15 ! Set point temp.
  double precision :: derT1, derT2           ! Temperature derivative
  double precision :: qCon1, qCon2           ! Heat Flux
  double precision, parameter :: Kp = 100    ! P Gain
  double precision, parameter :: h = 5       ! Convective heat transfer coefficient
  double precision, parameter :: G = 10      ! Conductivity
  double precision, parameter :: C = 10      ! Capacity

  integer :: i
  integer, parameter :: iun = 6              ! Logical unit number

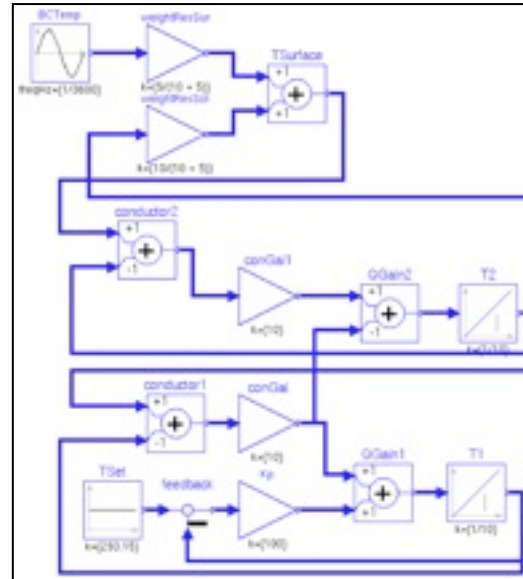
  double precision, parameter :: amp = 5     ! Amplitude
  double precision :: time                   ! Simulation time

  character(len=*) , parameter :: FMT = "(M14,7)"

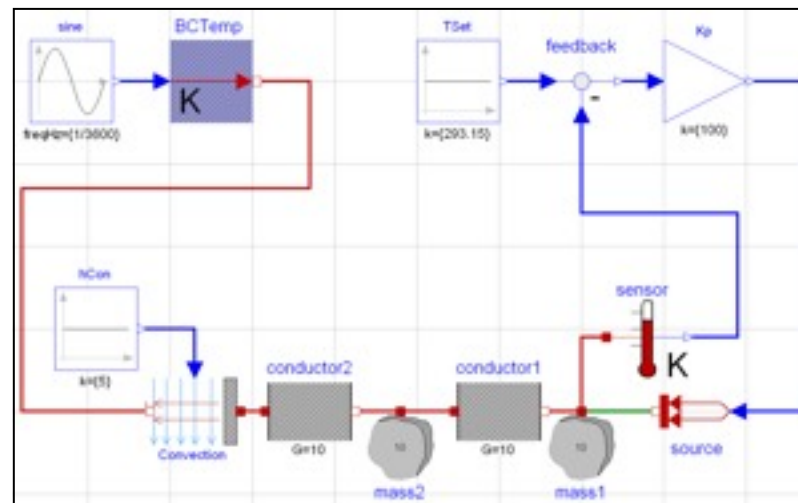
  ! Initialization variables
  dt = tFin/N
  T1 = T0
  T2 = T0
  time = 0
  iCon = 1

  open (iun, FILE='results.txt')
  ! Perform integration
  do i = 1, N, 1
    TBC = T0 + amp * sin(2*pi*i/hCon*(tFin/2000))
    TSur = T2 + (G / (h * C)) * TBC + T0 * (1 - G / (h * C))
    qCon1 = G * (TSur - T1)
    qCon2 = G * (TSur - T2)
    derT1 = 1/C * ( - qCon1 + qCon2 )
    derT2 = 1/C * ( qCon1 - qCon2 )
    if (i-hCon, iCon) then
      write(iun,FMT) time, T1, T2, qCon1
      iCon = iCon + hCon
    end if
    ! Update variables
    T1 = T1 + dt * derT1
    T2 = T2 + dt * derT2
    time = time + dt
  end do
  write(iun,FMT) time, T1, T2, qCon1
  close(iun)
  write(*,*) 'Program finished'
end program
  
```

Block diagram modeling  $\approx 1990$

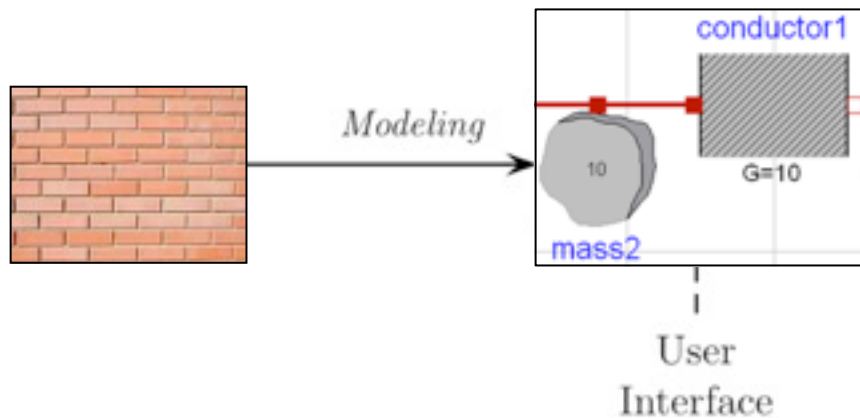


Equation-based, object-oriented modeling  $\approx 2000$



# Separation of Concerns

## Modeling



*Describes the phenomena*

- Standardized interfaces
- Acausal models
- Across & through variables
- Hierarchical modeling
- Class inheritance

## Compilation & Simulation

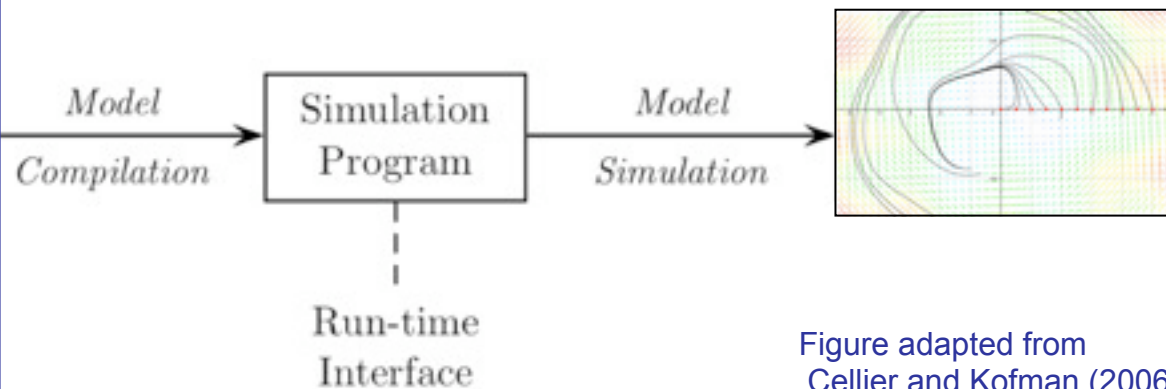
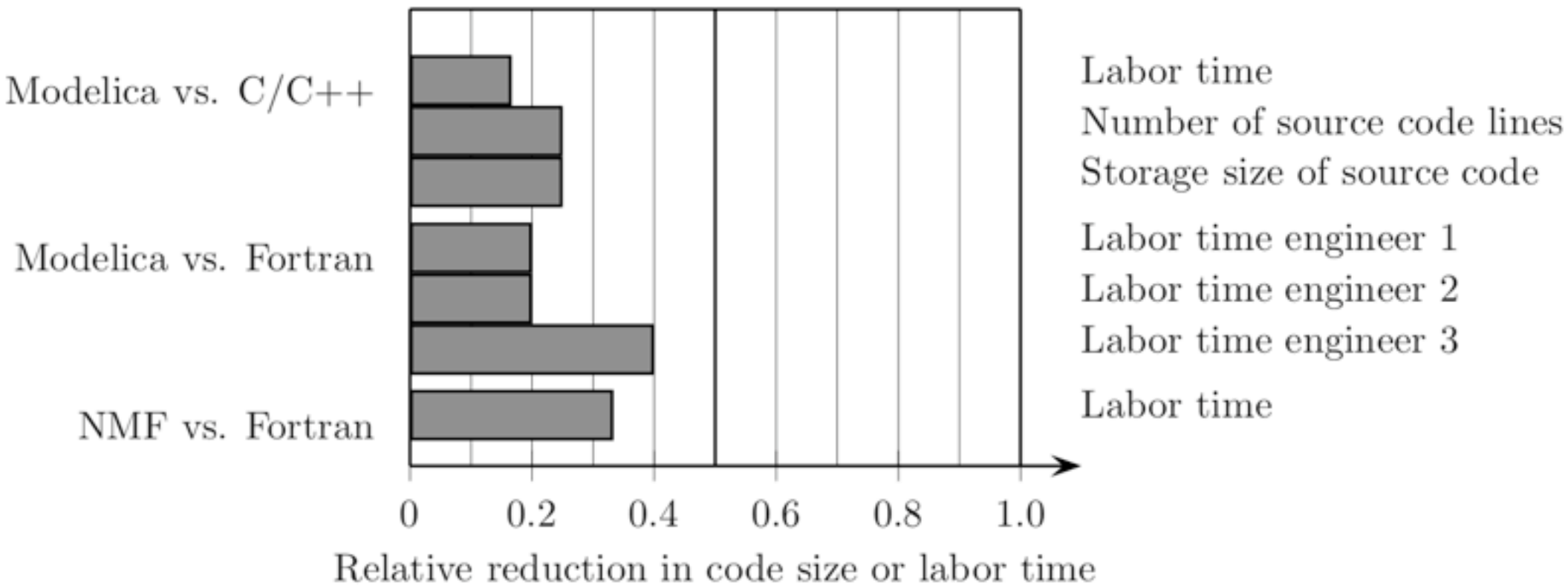


Figure adapted from  
Cellier and Kofman (2006)

*Solves the equations*

- Partitioning
- Tearing
- Inline integration
- Adaptive solver
  - Integration
  - Nonlinear equations

# Model Development Time



# Problems

- Building simulation programs are not designed for multi-disciplinary analysis
- Controls has wrong semantics
- Many modern building systems cannot be analyzed
- Adding models takes months
- Tools were not developed for
  - automated analysis
  - innovative systems
- Sharing models & data is hard
- Limited educational benefits due to black-box models and outdated technologies
- Heavy reliance on expensive and slow full scale experiments

# Overview

- Introduction

Trends - Problems - Needs

- Mono-Simulation with Modelica

Modelica Standard Library - LBNL Buildings Library - Applications

- Co-Simulation with Building Controls Virtual Test Bed

Analysis - Building Controls Virtual Test Bed - Applications

- R&D Needs

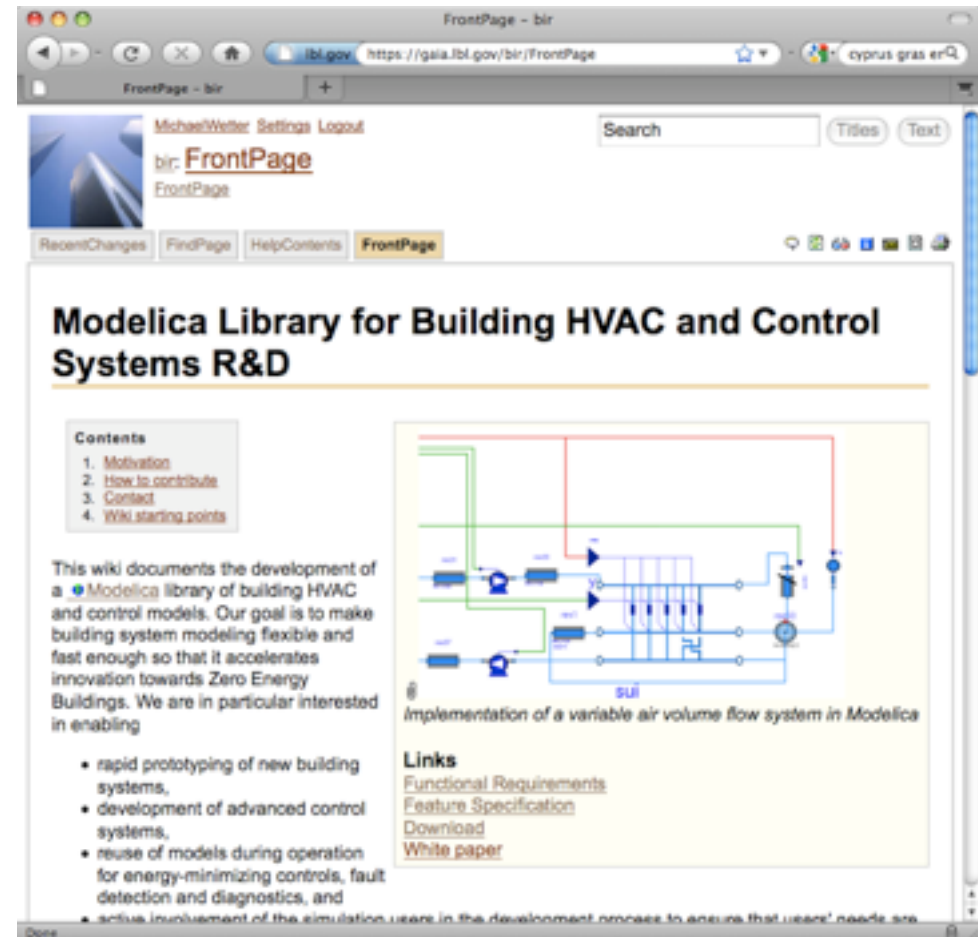
# Modelica Buildings Library

Enable

- Rapid prototyping of innovative systems
- Controls design
- Model-based operation

Available from

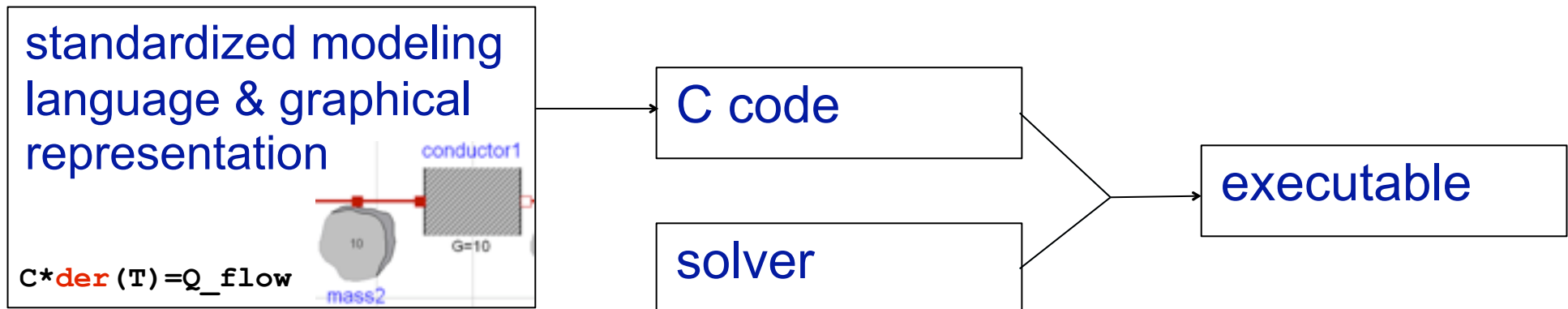
<http://simulationresearch.lbl.gov/modelica>



# What is Modelica

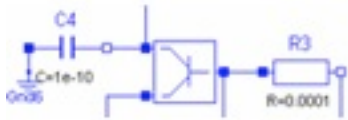


- Object-oriented equation-based language
- Icons with standardized interfaces encapsulate differential, algebraic and discrete equations
- Developed since 1996 because conventional approach for modeling was inadequate for integrated engineered systems
- **Well positioned to become de-facto open standard** for modeling multi-engineering systems
  - ITEA2: 285 person years investment over next three years.

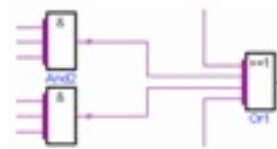


# Modelica Standard Library. 1300 models & functions.

## Analog



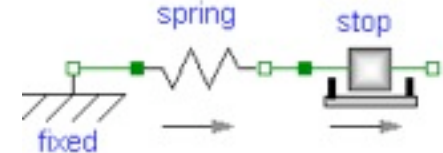
## Digital



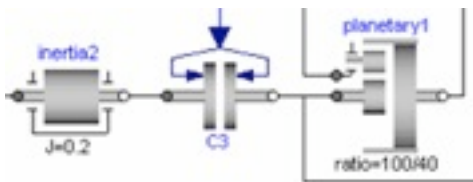
## Machines



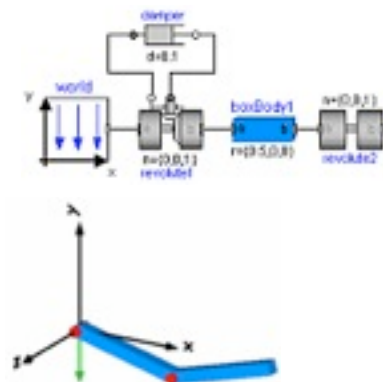
## Translational



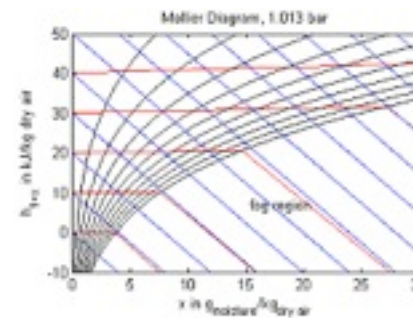
## Rotational



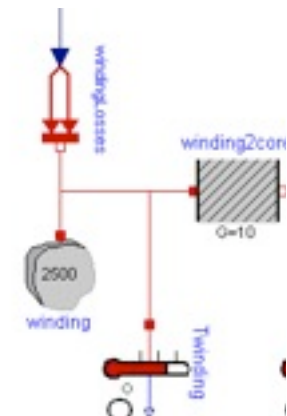
## MultiBody



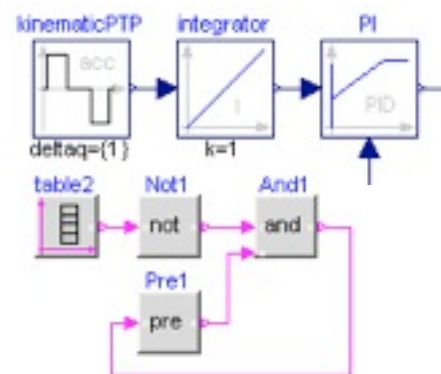
## Media



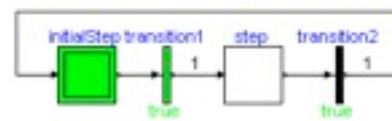
## HeatTransfer



## Blocks



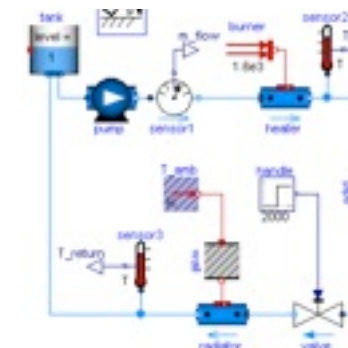
## StateGraph



## Math

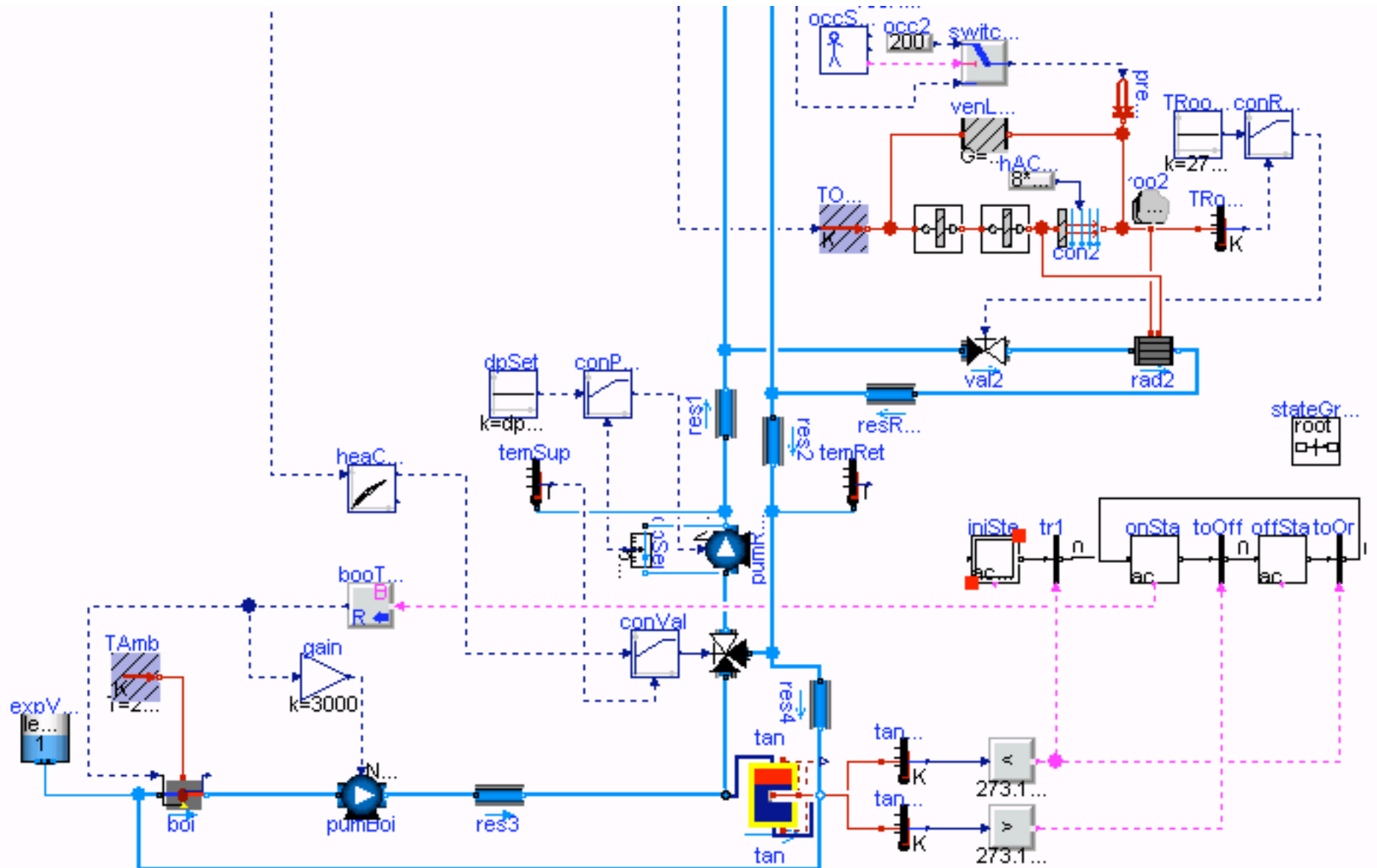
$$\|v\|_p = \left( \sum_{i=1}^n |v[i]|^p \right)^{1/p}, \quad 1 \leq p \leq \infty$$

## Fluid



# LBNL Buildings Library. 100 models and functions.

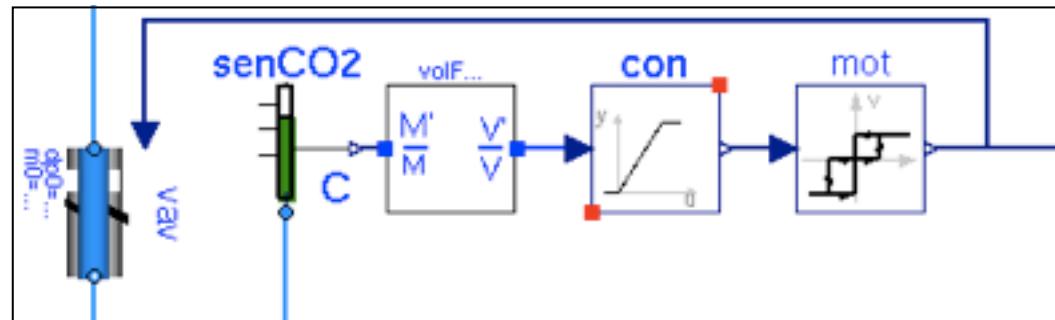
Provides HVAC specific models based on Modelica.Fluids library



# Usage Levels

## Model user

Drags & drops component models to form system model



## Model developer

Reuses base models to implement new models

```
Q_flow = Q0_flow * u;  
mXi_flow = zeros(Medium.nXi);
```

## Library developer

Develops base models for characteristic components

```
port_a.m_flow*port_b.h_outflow +  
    port_b.m_flow*inStream(port_a.h_outflow) = Q_flow;  
port_a.m_flow + port_b.m_flow = -sum(mXi_flow);
```

# Applications

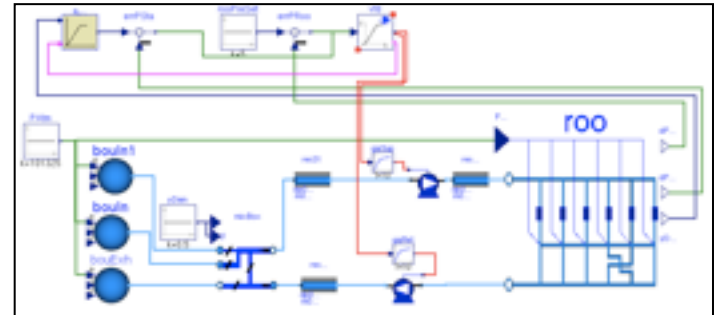
## 1) Rapid prototyping

Analyzed novel hydronic heating system with radiator pumps and hierarchical system controls.



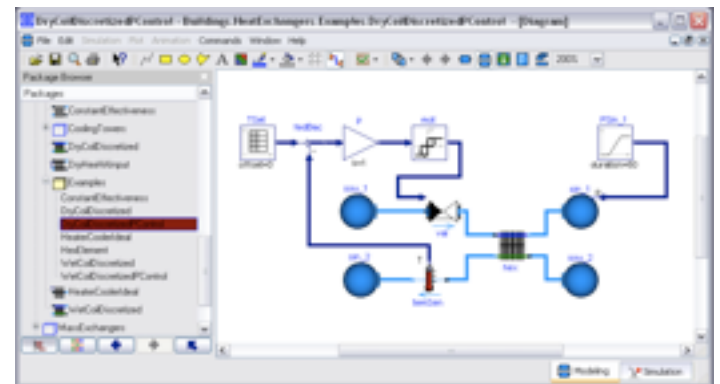
## 2) Supervisory controls

Simulated & auto-tuned “trim and response” sequence for variable air volume flow systems.

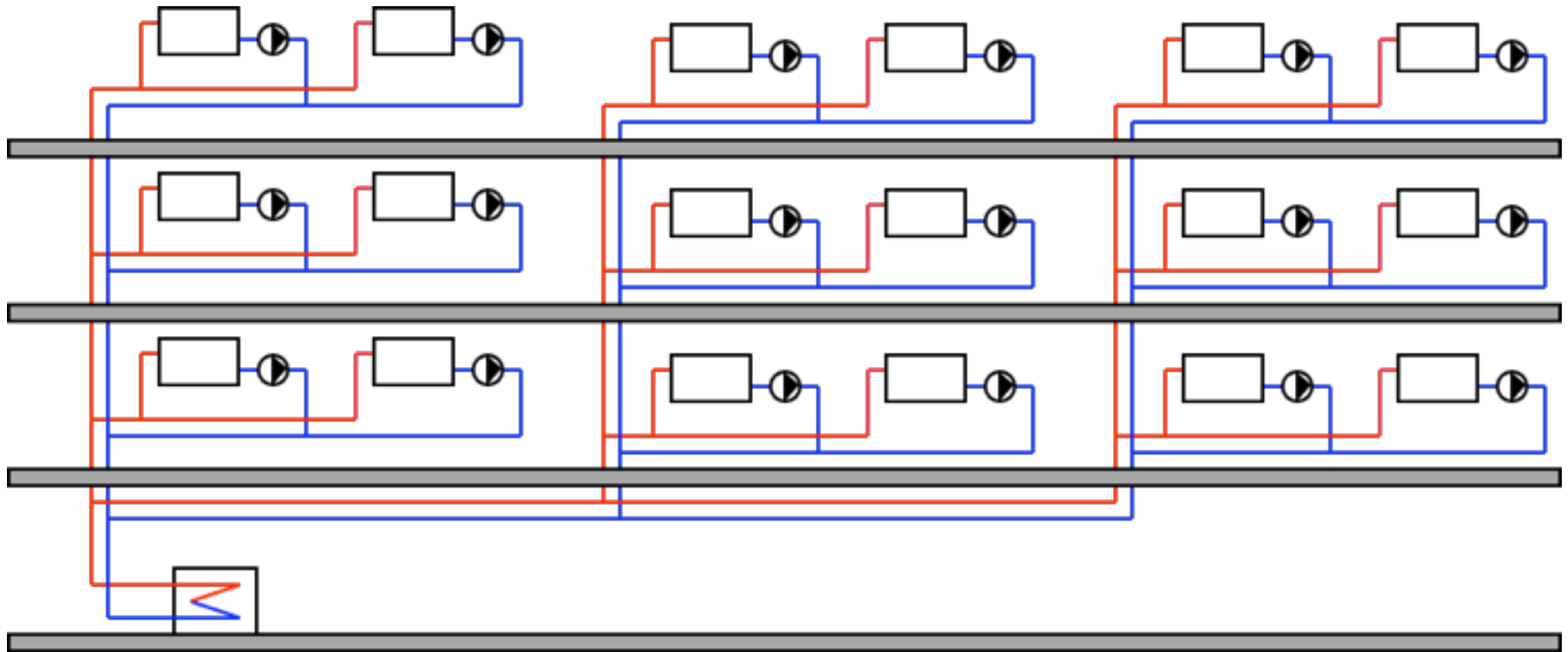


## 3) Local loop controls

Reused high-order model for controls design in frequency domain.



# Rapid Prototyping: Wilo GENIAX (Introduction 2009)



## Original system model

2400 components

13,200 equations

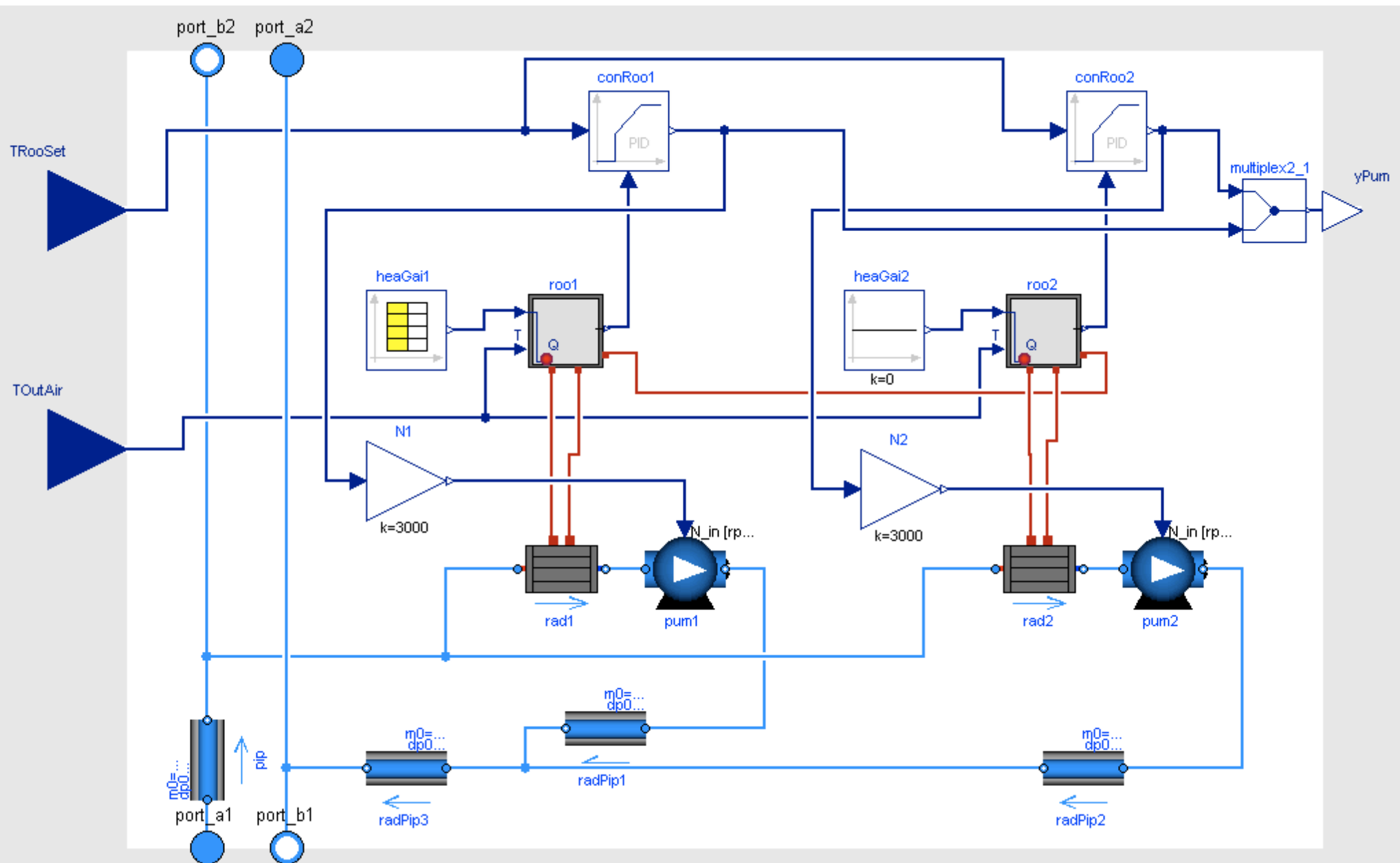
## After symbolic manipulations

300 state variables

8,700 equations

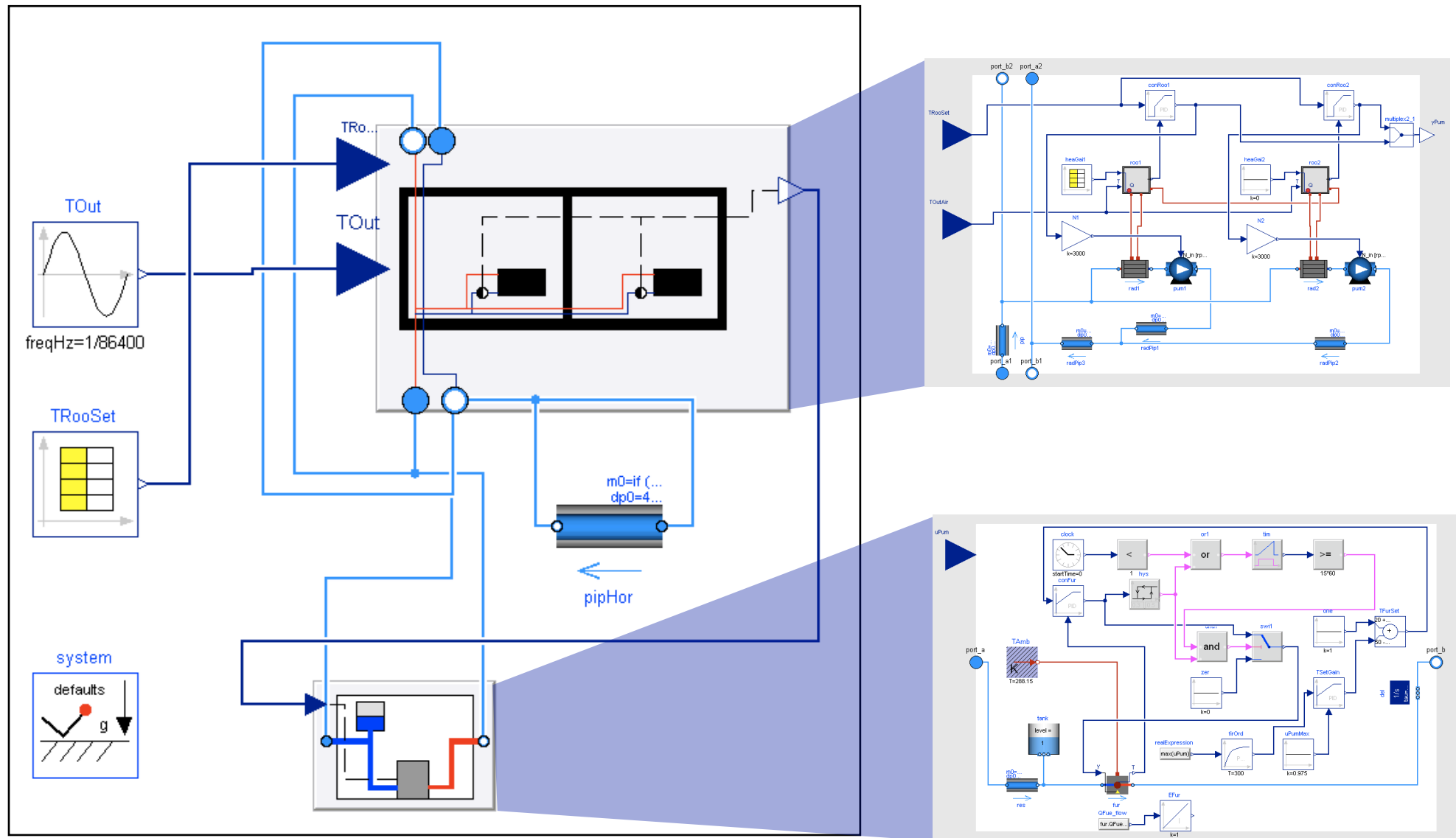


# Rapid Prototyping: Wilo GENIAX (Introduction 2009)



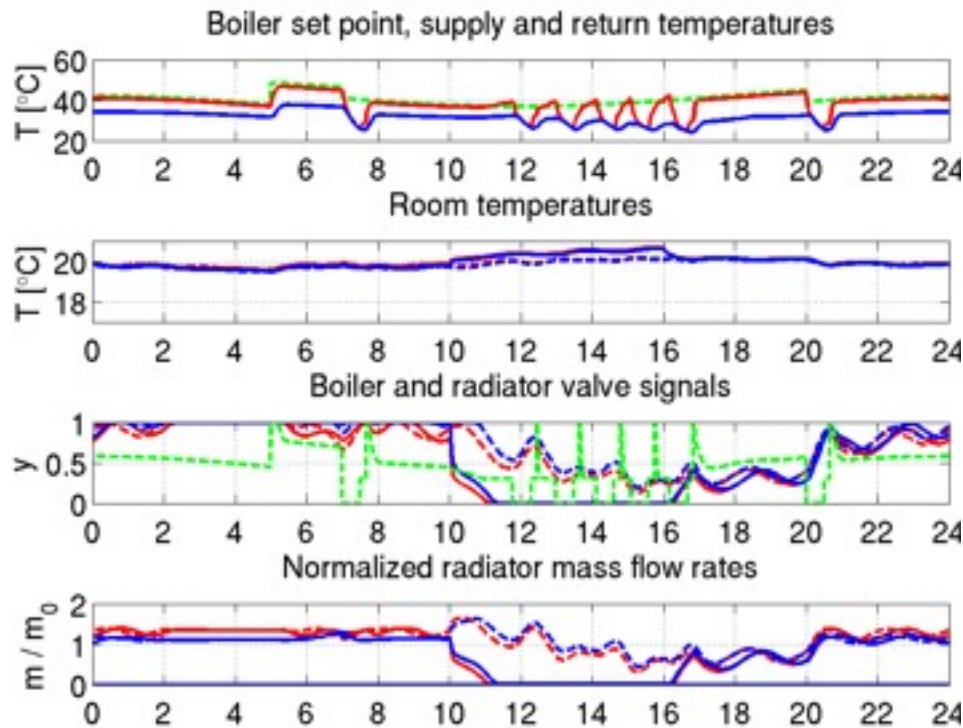
[illegible]

# Rapid Prototyping: Wilo GENIAX (Introduction 2009)

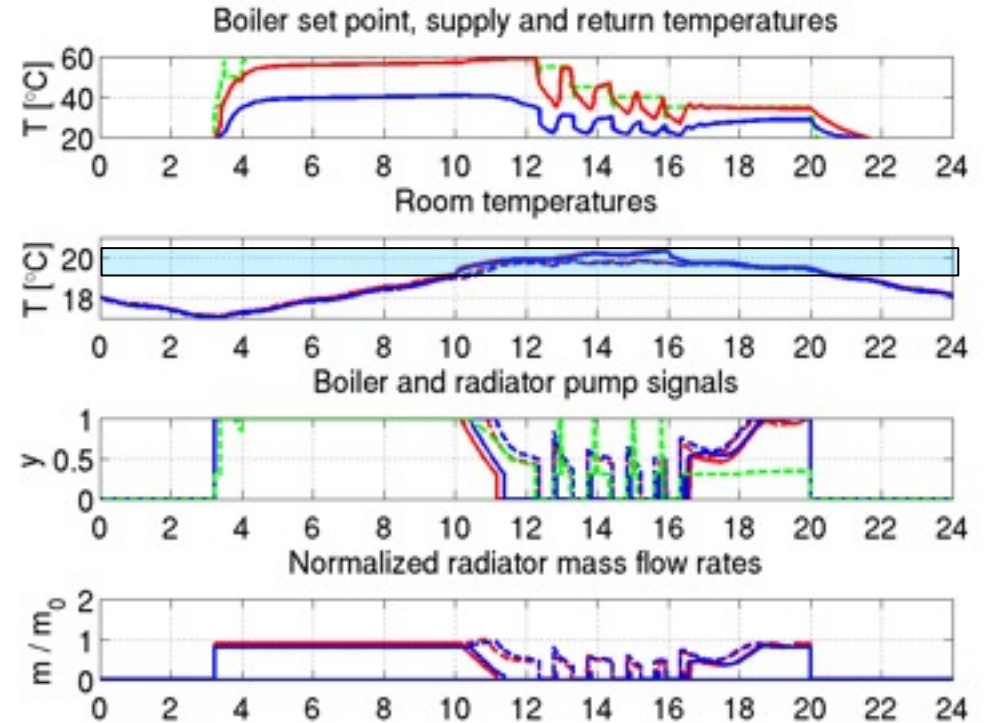


# Rapid Prototyping: Wilo GENIAX (Introduction 2009)

## Thermostatic radiator valves



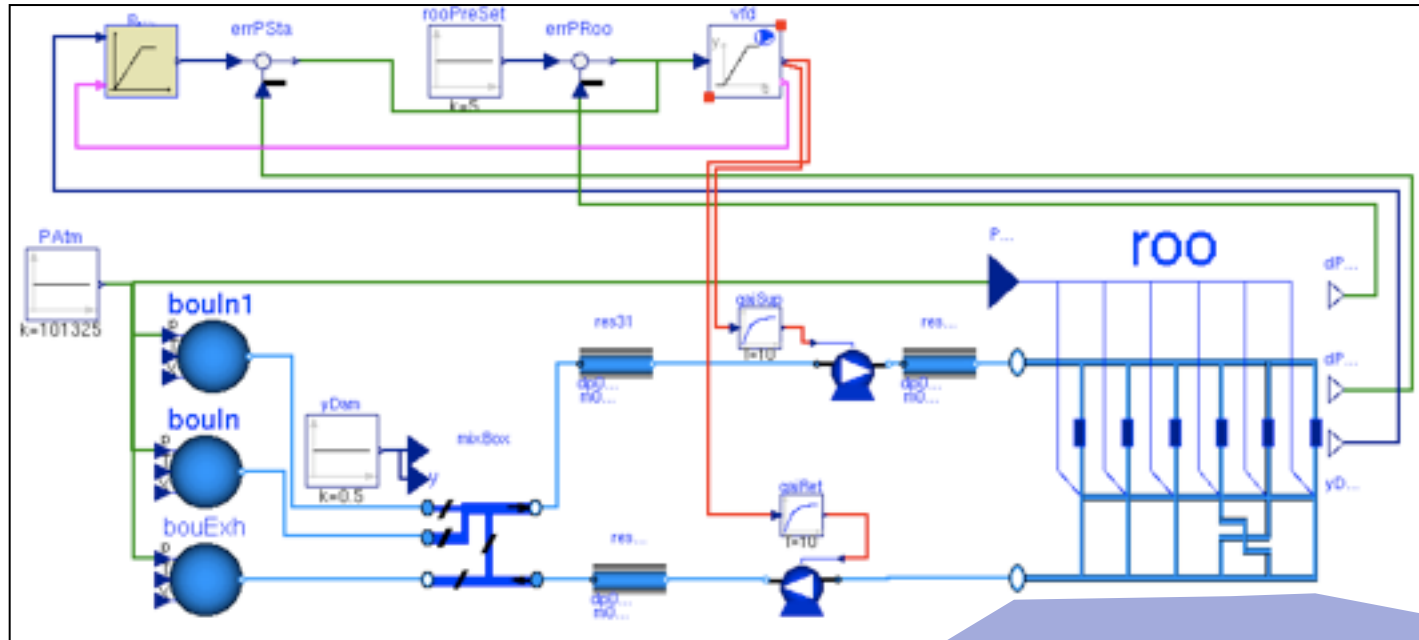
## Radiator pumps



Reproduced trends published by Wilo.

Developed boiler model, radiator model, simple room model and both system models in one week.

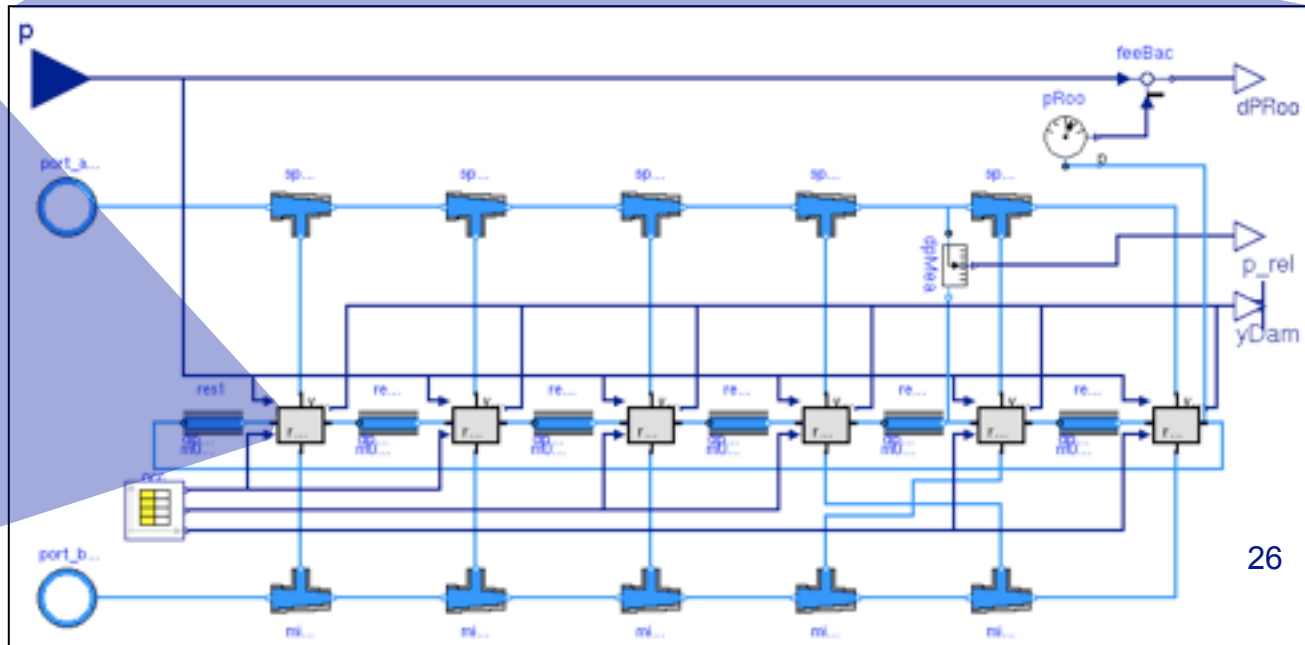
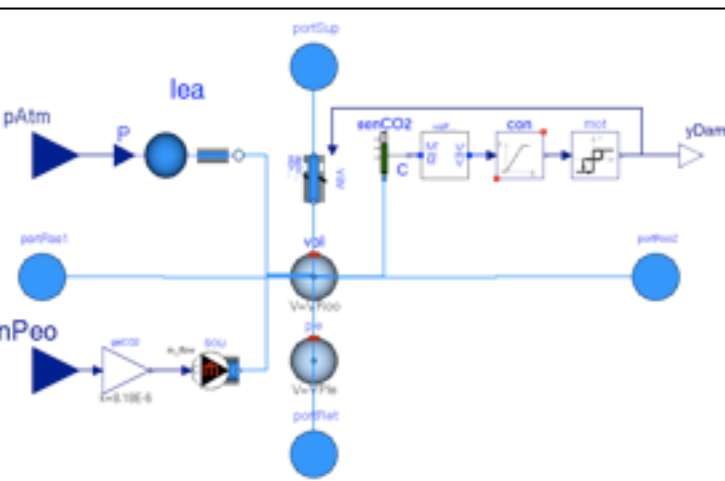
# Applications – VAV System Controls



VAV System  
(ASHRAE 825-RP)

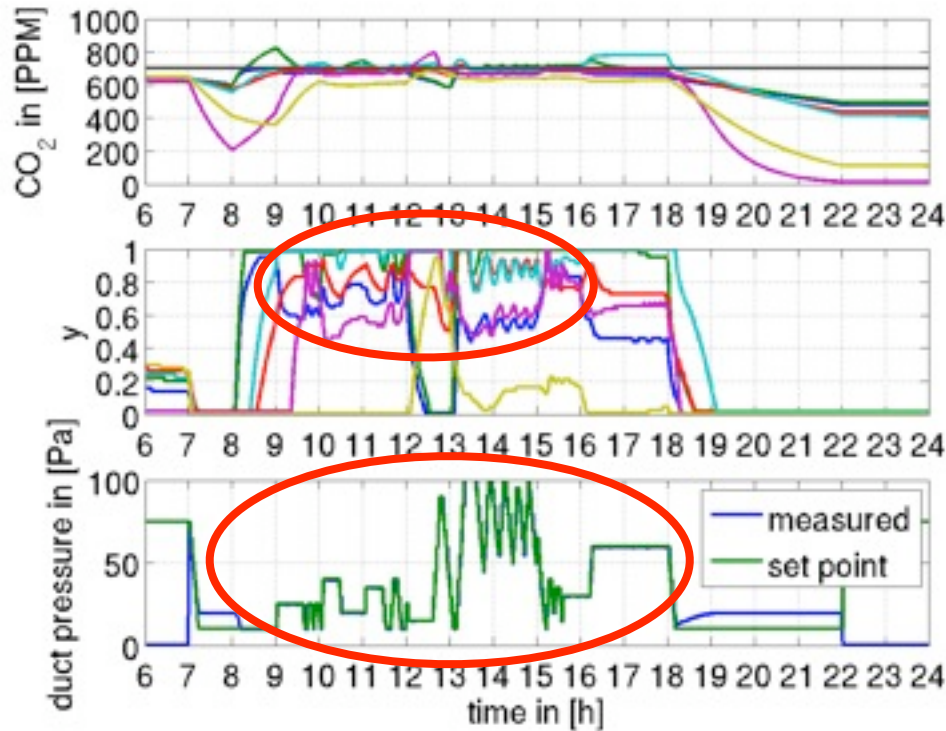
Trim & response control for fan  
static pressure reset  
(Taylor, 2007)

**Original system model**  
730 components  
4,420 equations  
40 state variables



# Applications – VAV System Controls

Stabilized control and reduced energy by solving optimization problem with state constraints

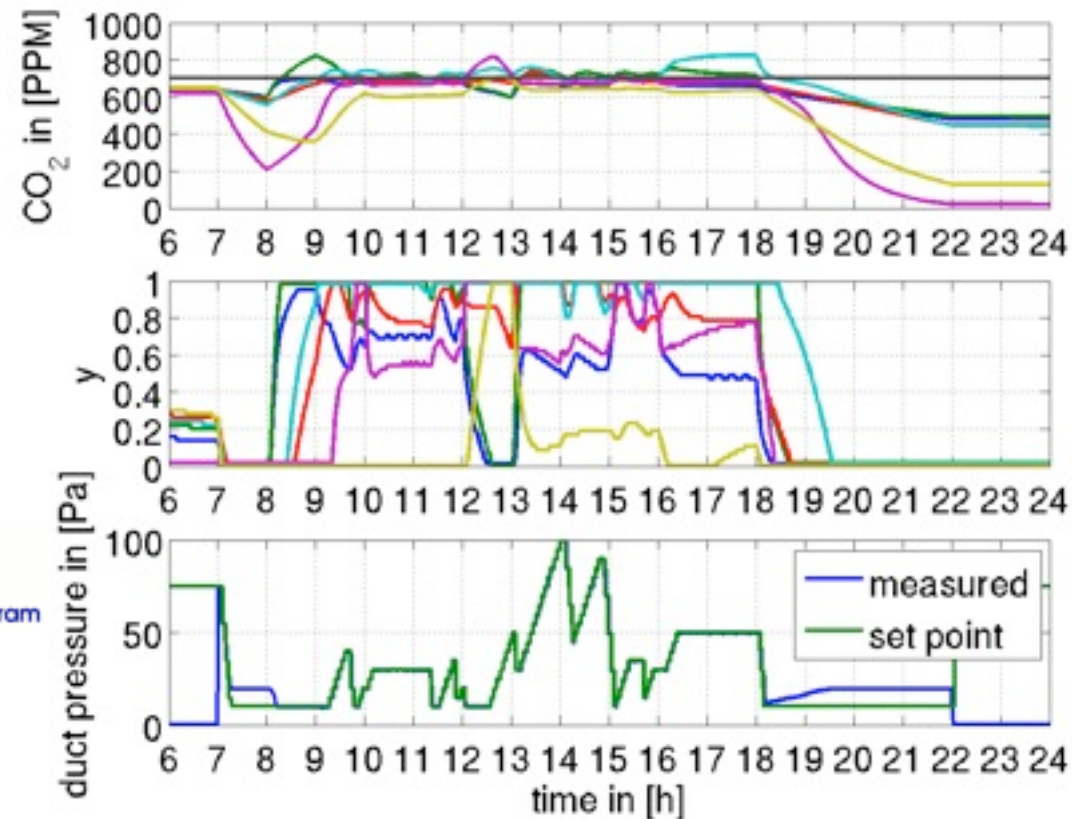


$$\min_{x \in \mathbf{X}} \{f(x) \mid g(x) = 0\},$$

$$f(x) = \frac{1}{E_0} \int_0^T P_f(x, t) dt,$$

$$g(x) = \frac{1}{T} \int_0^T \left( \max\{0, (y_j(x, t)/\hat{x}_s) - 1/(2 K_p) - 1 \mid j \in \mathbf{J}(x, t)\} \right)^2 dt$$

**GenOpt**  
Generic Optimization Program



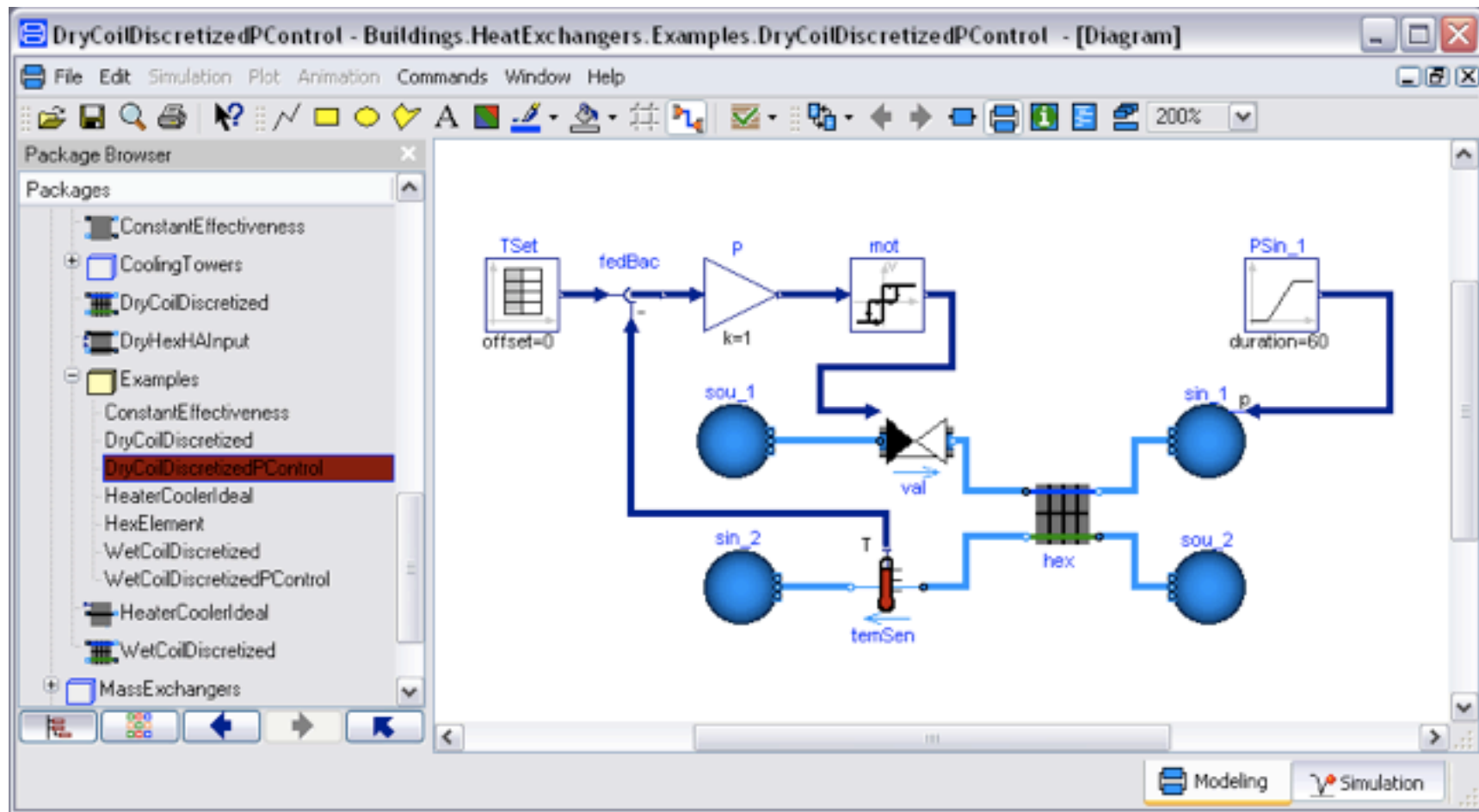
# Applications – Feedback Loop Stability

## Heat exchanger feedback control

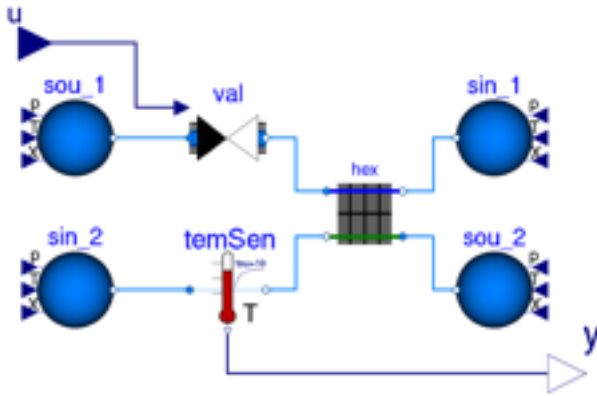
2632 equations

40 states

37x37 (linear) + 6x6 (nonlinear)  $\longrightarrow$  0 (linear) + 2x2 (nonlinear)



# Applications – Feedback Loop Stability

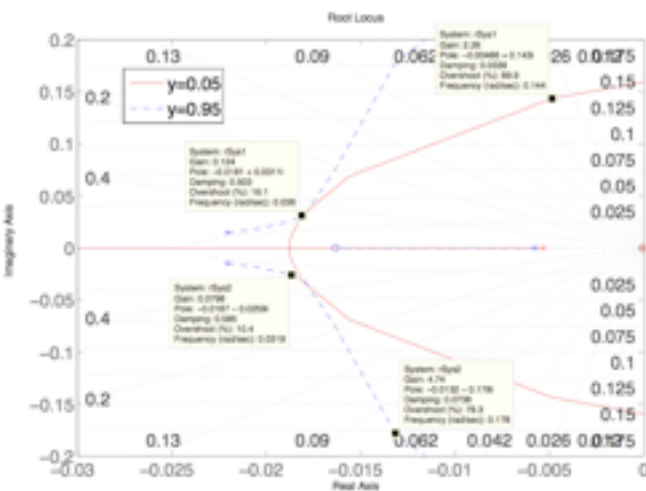
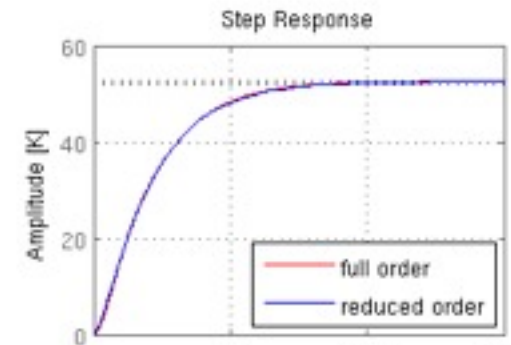
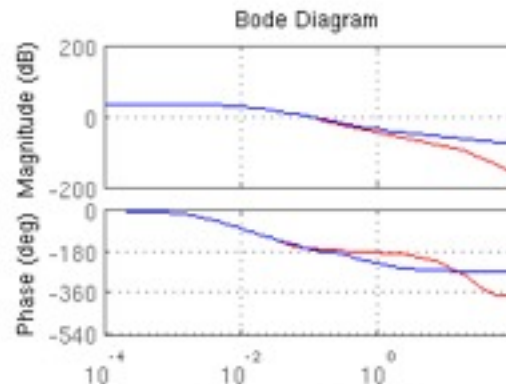


$$\dot{x}(t) = A x(t) + B u(t)$$

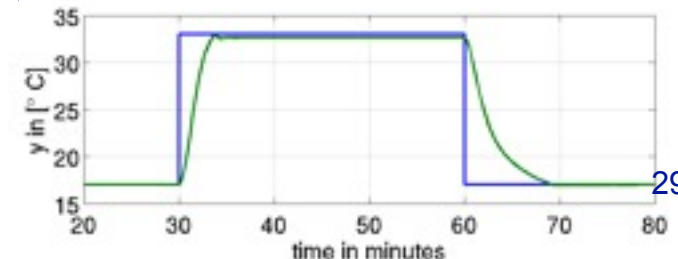
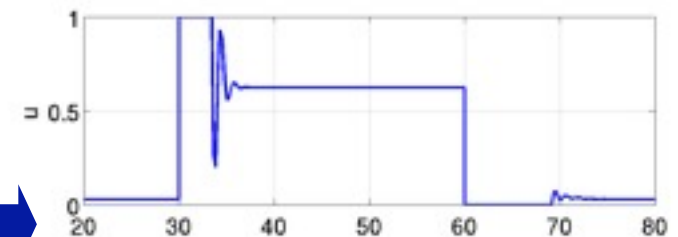
$$x(t) \in \mathbb{R}^{40}$$

$$\dot{\tilde{x}}(t) = \tilde{A} \tilde{x}(t) + \tilde{B} \tilde{u}(t)$$

$$\tilde{x}(t) \in \mathbb{R}^3$$



$$u(t) = K(y) e(t)$$



# Overview

- Introduction

Trends - Problems - Needs

- Mono-Simulation with Modelica

Modelica Standard Library - LBNL Buildings Library - Applications

- **Co-Simulation with Building Controls Virtual Test Bed**

Analysis - Building Controls Virtual Test Bed - Applications

- **R&D Needs**

# Building Controls Virtual Test Bed (BCVTB)

## Enable

- Co-simulation for integrated multi-disciplinary analysis
- Use of domain-specific tools
- Model-based system-level design
- Model-based operation

## Available from

<http://simulationresearch.lbl.gov/bcvtb>

Based on Ptolemy II from UC Berkeley, which will include BCVTB interface.

The screenshot shows a web browser window titled "FrontPage - bcvtb" with the URL "https://gaia.lbl.gov/bcvtb/FrontPage". The page features a navigation bar with links like "RecentChanges", "FindPage", "HelpContents", and "FrontPage". The main content area is titled "Building Controls Virtual Test Bed" and contains a detailed description of the BCVTB software environment. It mentions that BCVTB is based on Ptolemy II and allows for co-simulation of building and HVAC systems with EnergyPlus and control logic in MATLAB/Simulink. A diagram on the right illustrates the system architecture, showing the flow of data between EnergyPlus, Simulink, and other components. Below the diagram, there are links to "Simple application for illustration", "Implementation", "Getting started", "Development", and "Help". At the bottom, a section titled "Links" provides further resources.

The Building Controls Virtual Test Bed (BCVTB) is a software environment that allows expert users to couple different simulation programs for distributed simulation. For example, the BCVTB allows to simulate a building and HVAC system in EnergyPlus and the control logic in MATLAB/Simulink, while exchanging data between the software as they simulate. The BCVTB is based on the Ptolemy II software environment. The BCVTB is still under development and aimed at expert users of simulation. Due to the different programs that may be involved in distributed simulation, familiarity with compiling and configuring programs is essential.

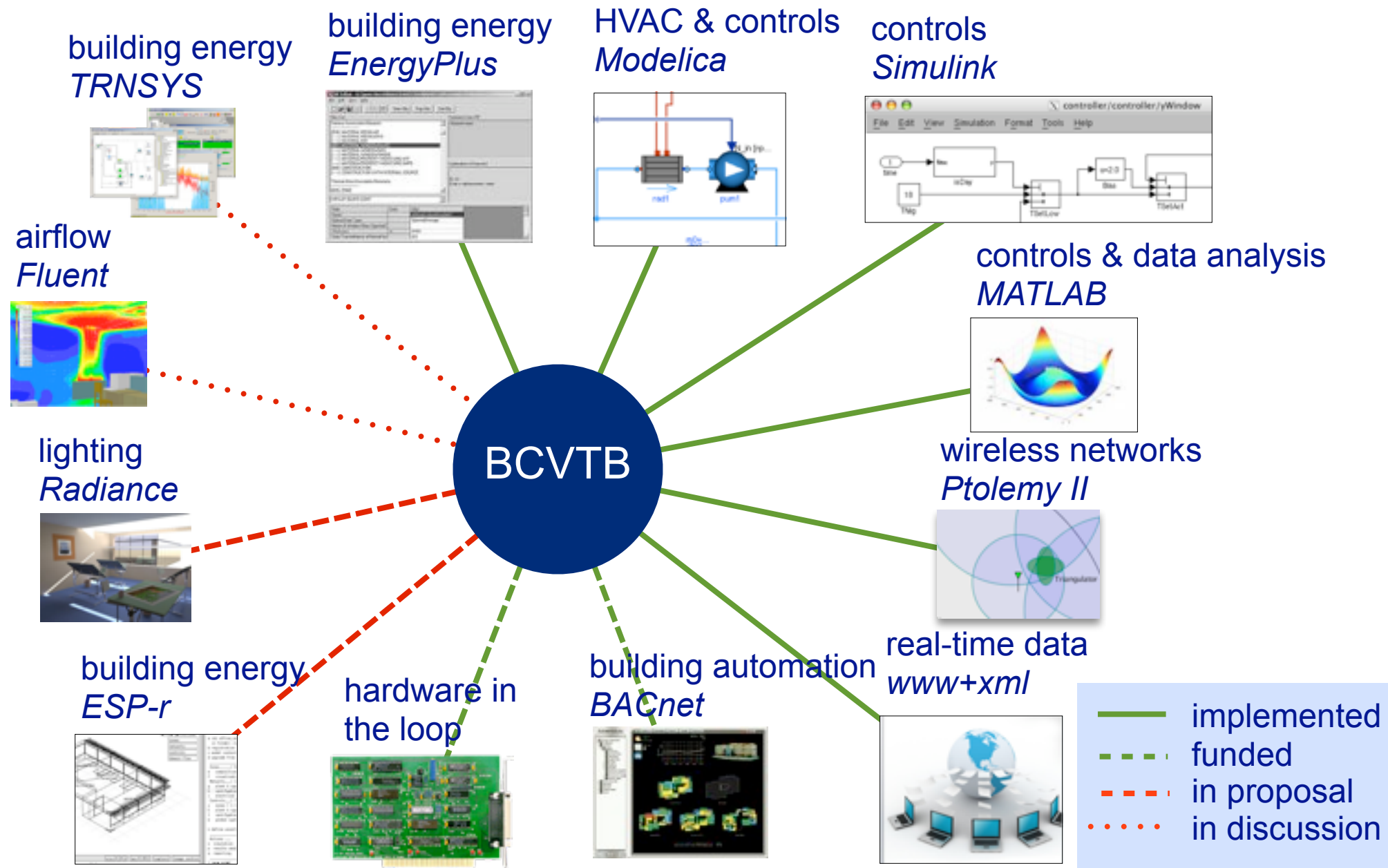
Programs that are linked to the BCVTB are

- EnergyPlus.
- MATLAB.
- Simulink and
- Dymola, which is a Modelica modeling and simulation environment.

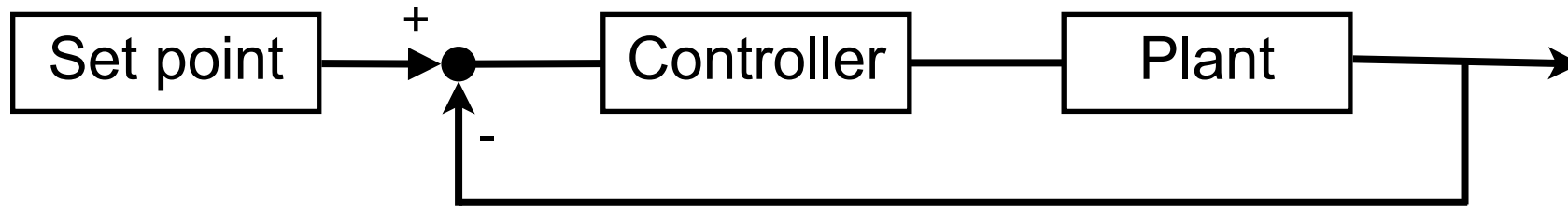
In future work we will link a BACnet compliant Building Automation System (BAS) and digital/analog converters to the BCVTB. In addition to using programs that are coupled to Ptolemy II, Ptolemy II's graphical modeling environment can also be used to define models for control systems, for physical devices, for communication systems or for post-processing and real-time visualization.

# Functional Domains & Coupled Tools

Middle-ware that exchanges and synchronizes data as (simulation-)time progresses



# Simple Example: Room Heater



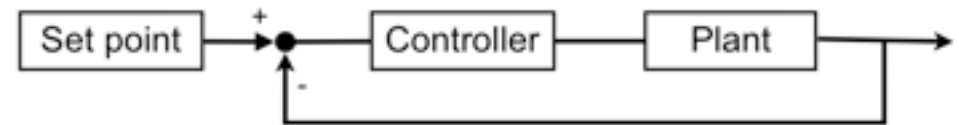
**Controller: Discrete time proportional controller**

$$y(k+1) = \max(0, \min(1, K_p (T_{set} - T(k))))$$

**Plant: Room model**

$$T(k+1) = T(k) + \frac{\Delta t}{C} \left( U A (T_{out} - T(k)) + \dot{Q}_0 y(k) \right)$$

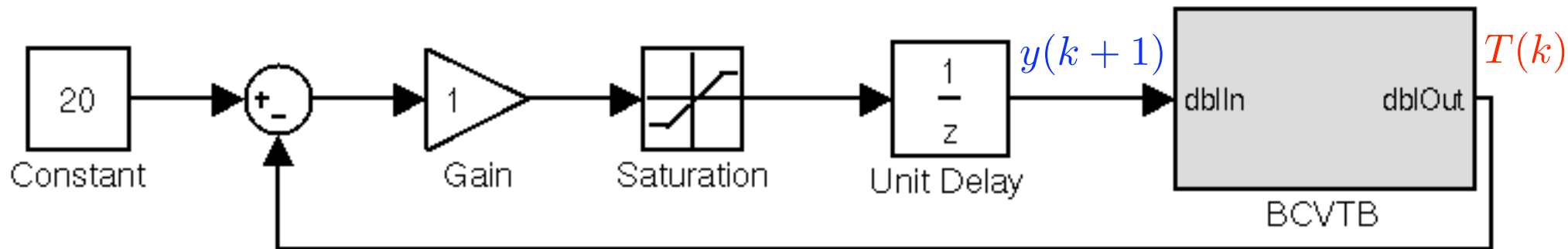
# Simple Example: Room Heater



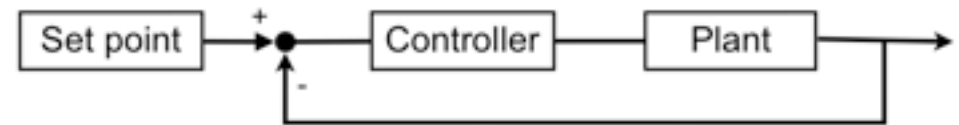
## Discrete Time Proportional Controller

$$y(k+1) = \max(0, \min(1, K_p (T_{set} - T(k))))$$

### Implementation in Simulink



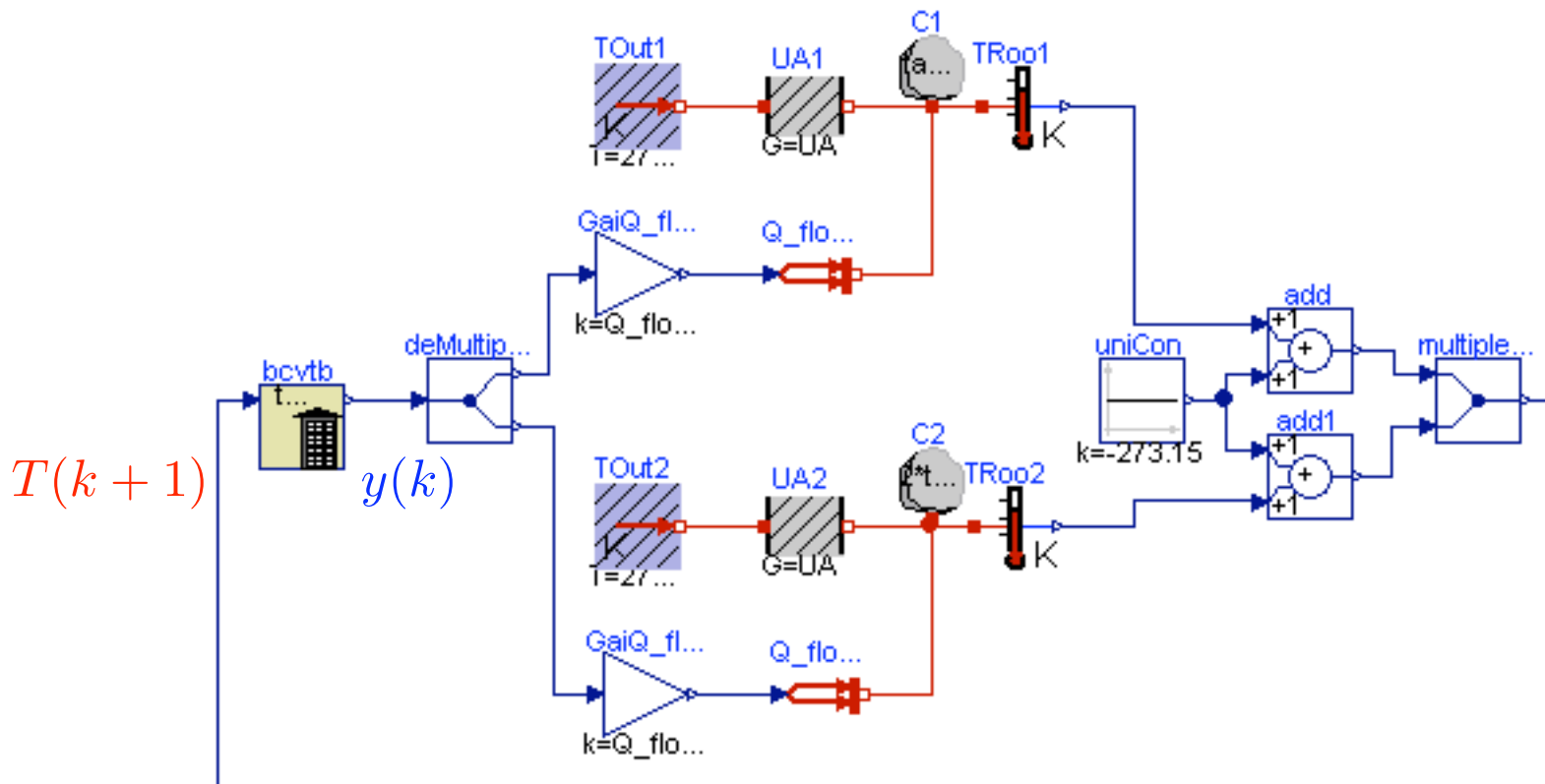
# Simple Example: Room Heater



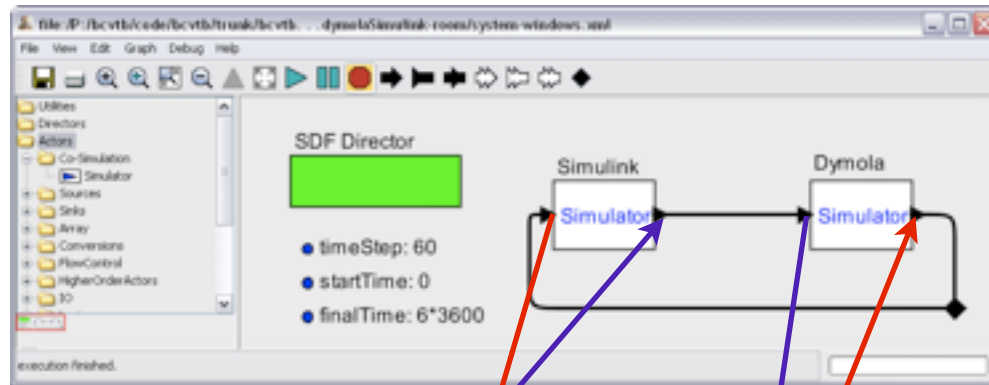
## Room model

$$T(k+1) = T(k) + \frac{\Delta t}{C} \left( UA (T_{out} - T(k)) + \dot{Q}_0 y(k) \right)$$

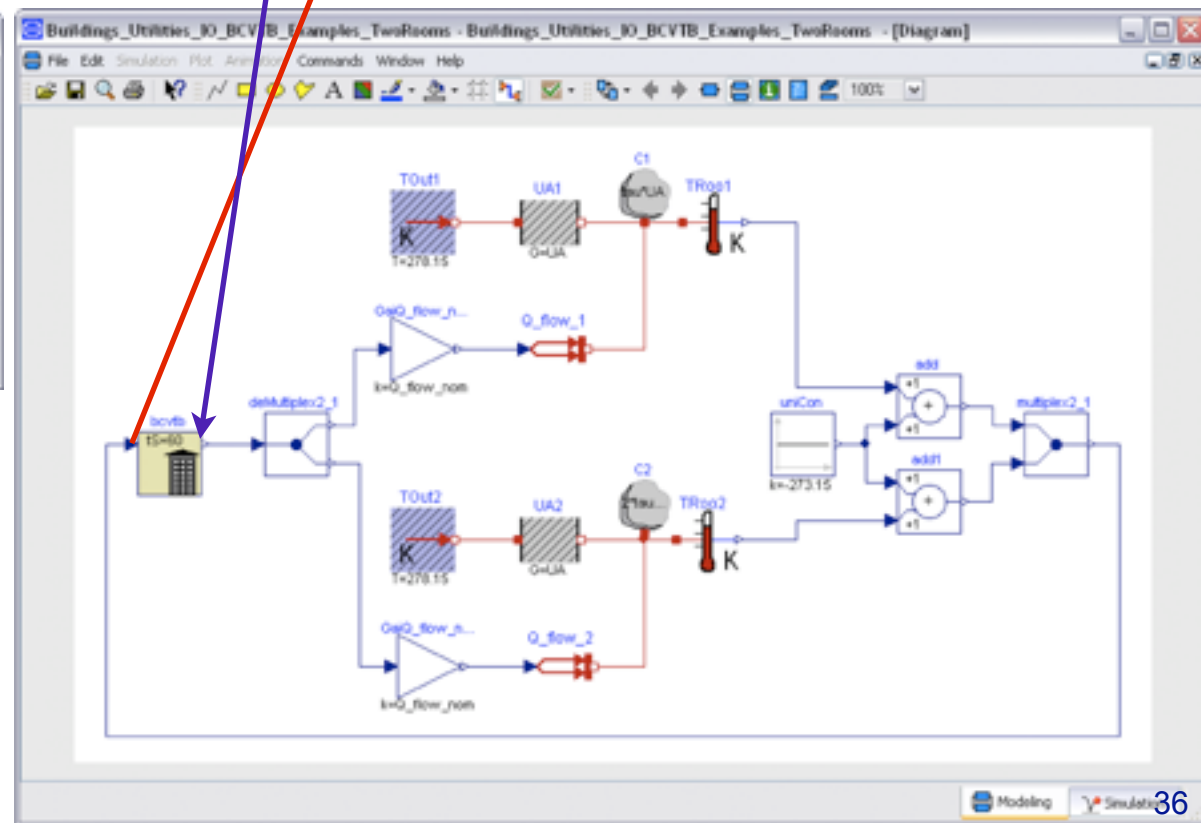
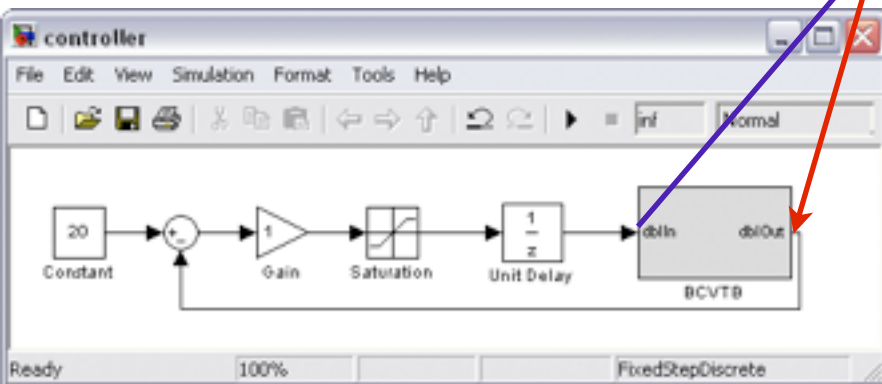
## Implementation in Modelica



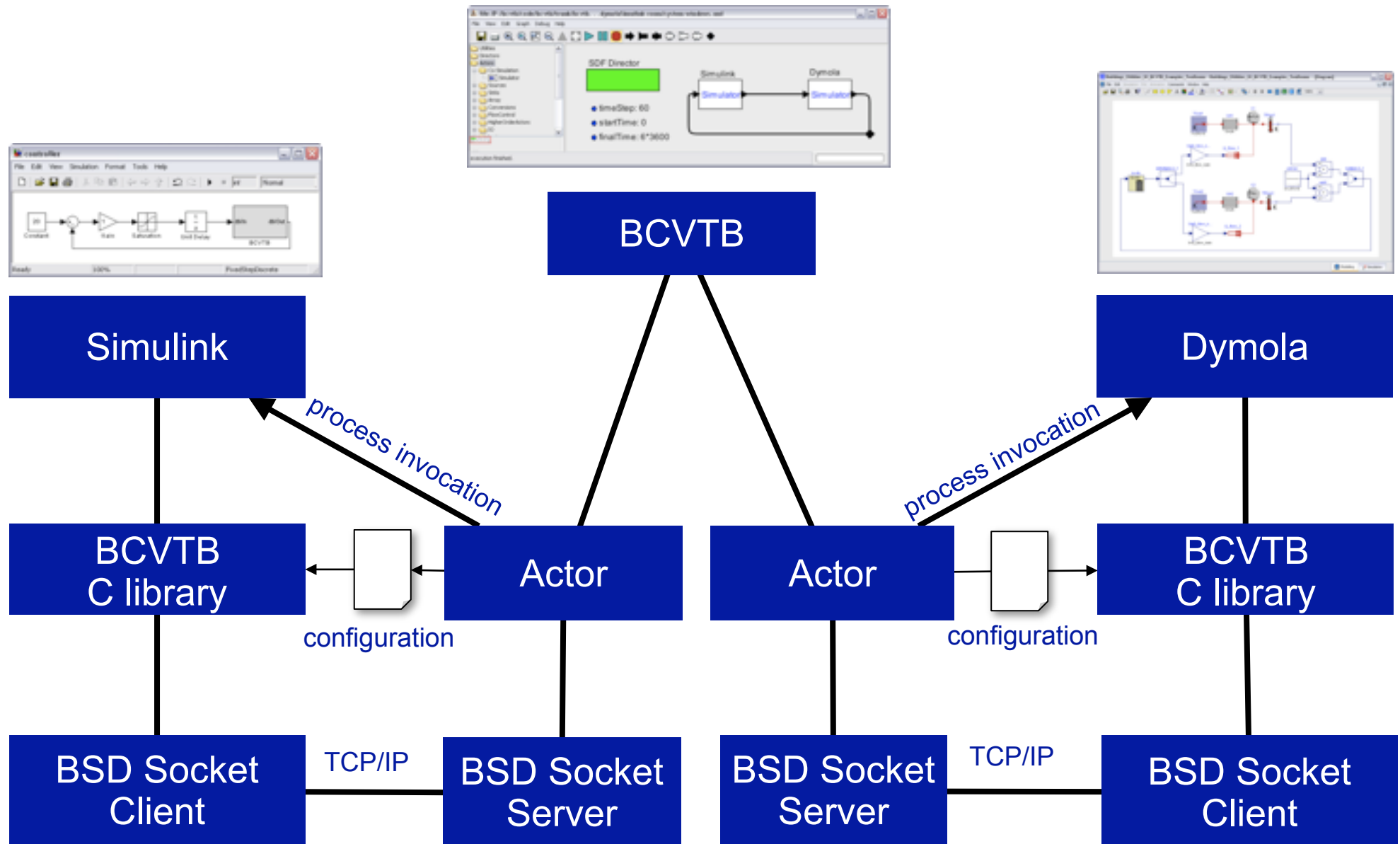
# Simple Example: Room Heater



$y(k+1)$   $T(k)$   $y(k)$   $T(k+1)$



# BCVTB Architecture



# Co-Simulation Analysis

- 1) Does numerical solution of co-simulation **converge** to solution of differential equation?
- 2) How does exchanged data affect **stability**?
- 3) Is strong coupling or loose coupling more **efficient**?

# Co-Simulation Analysis

**Does numerical solution of co-simulation converge to solution of differential equation?**

*Consistency + stability = convergence*

## Consistency

a) *Definition:* Local Truncation Error, LTE = error produced in one integration step (starting from exact solution)

b) *Definition:* Unit Local Truncation Error,

$$\text{ULTE}(\Delta t) = \frac{\text{LTE}(\Delta t)}{\Delta t}$$

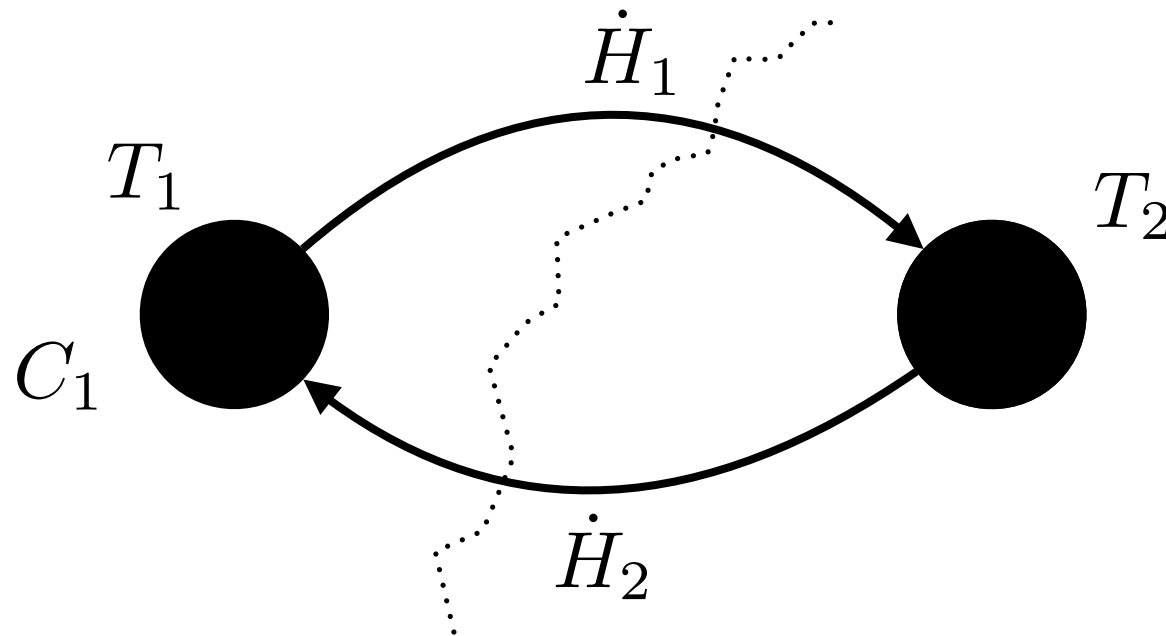
c) *Theorem:*  $\|\text{ULTE}_p(\Delta t)\| \leq \|\text{ULTE}(\Delta t)\| + \alpha L \Delta t$

## Stability

d) *Theorem:* Co-simulation is (conditionally) stable.

# Co-Simulation Analysis

How does exchanged data affect stability?



$$T_1(t^{n+1}) - T_1(t^n) \propto \frac{\int_{t^n}^{t^{n+1}} (\dot{H}_2(s) - \dot{H}_1(s)) ds}{C_1}$$

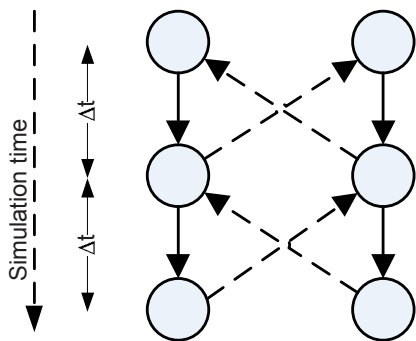
Couple to states with large capacity.

Instability typically happens after setpoint changes.

# Co-Simulation Analysis

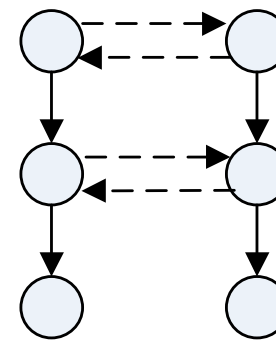
Is strong coupling or loose coupling more efficient?

*Strong coupling*

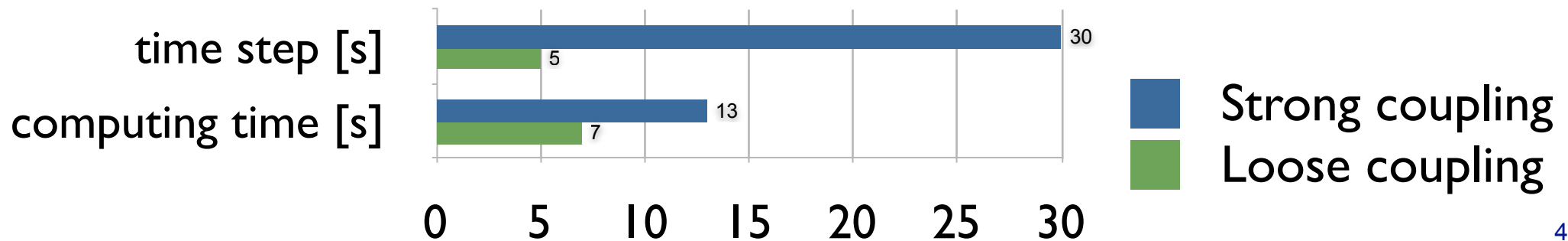


Requires iteration across simulators.  
Requires rewinding states.

*Loose coupling*

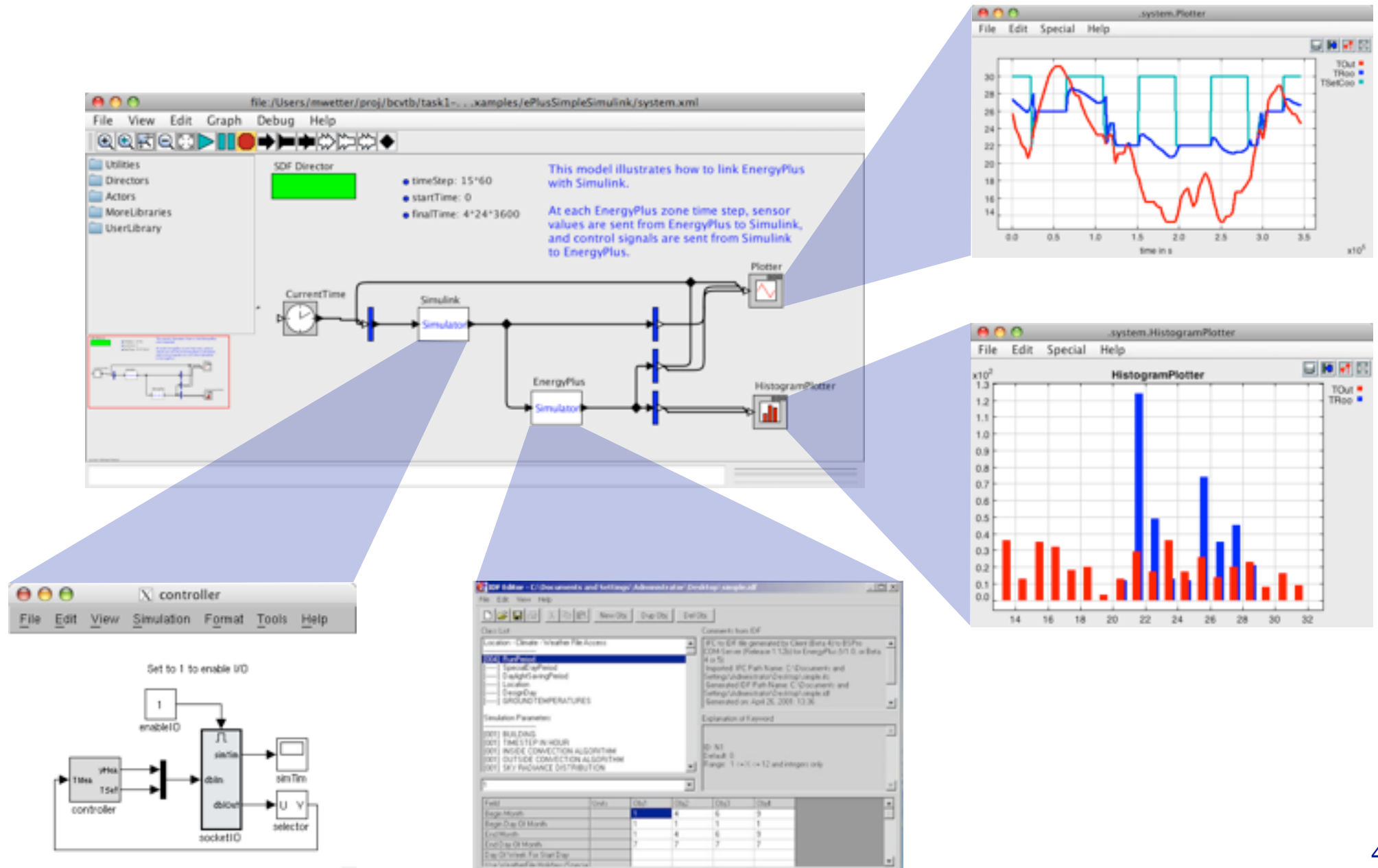


No iteration across simulators.  
Requires smaller time steps.



Details: See Trcka, Wetter, Hensen 2007

# Ex: Controls in Simulink, Building in EnergyPlus



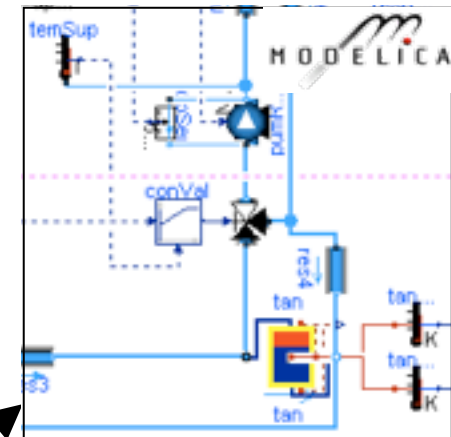
# Demonstration

# Reusable modules for model-based operation

Tool selection depends on required

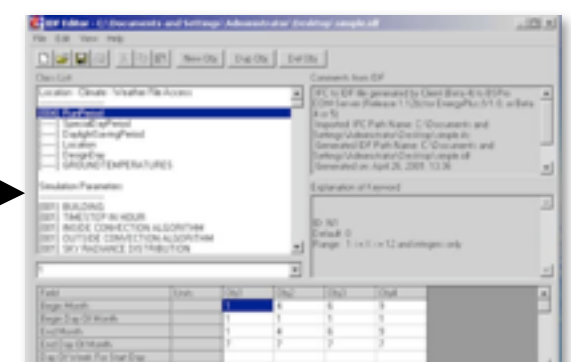
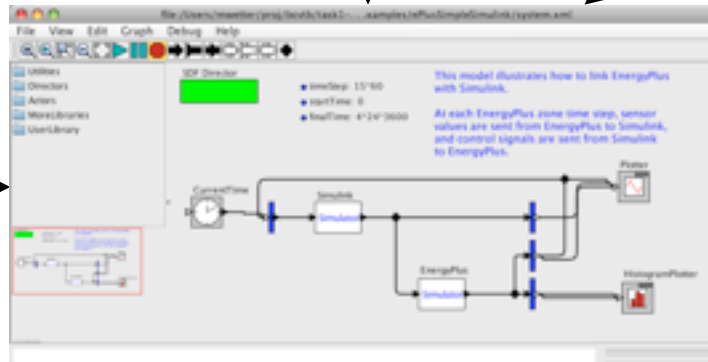
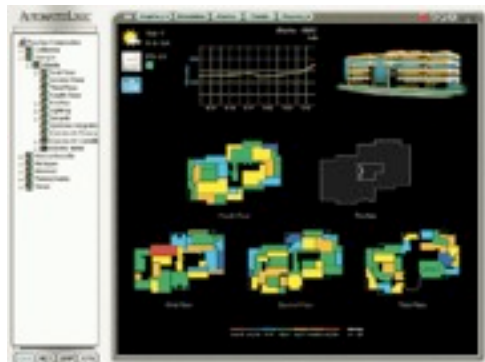
- model resolution
  - emulation of actual control & operation
  - dynamics of equipment
- analysis capabilities
  - smoothness
  - state initialization

Hybrid systems,  
emulate actual  
feedback control



Discrete time,  
idealized controls

www/xml



# Overview

- Introduction

Trends - Problems - Needs

- Mono-Simulation with Modelica

Modelica Standard Library - LBNL Buildings Library - Applications

- Co-Simulation with Building Controls Virtual Test Bed

Analysis - Building Controls Virtual Test Bed - Applications

- R&D Needs

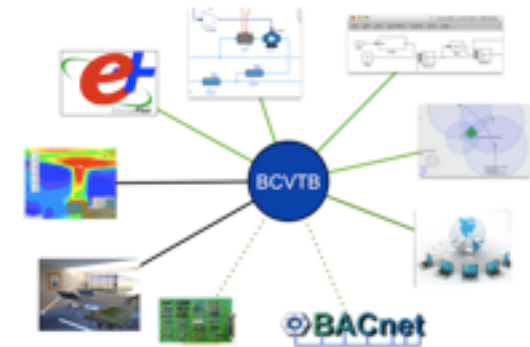
# R&D Needs

## Integration across disciplines

- model-based, system-level design processes
- design for robustness

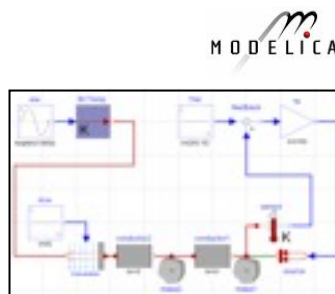
## Co-simulation

- adaptive step size
- semantics of exchanged data
- standardized data exchange
- distributed computing



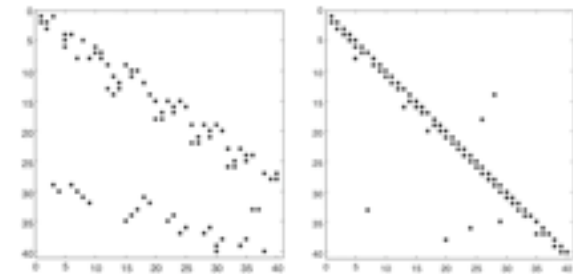
## Equation-based object-oriented modeling

- standardized libraries
- computationally efficient and robust model formulation
- code generation for controls



## Equation-based object-oriented simulation

- multi-rate solvers
- mapping equations to parallel hardware



## Optimization



- integration with design tools
- parallel algorithms (with cloud computing)
- stochastic optimization

*Downloads and further information:*  
<http://simulationresearch.lbl.gov/wetter>